(19) [logo] European Patent Office (11) Publication no.: 0 576 113 A2

(12) EUROPEAN PATENT APPLICATION

(21) Application no.: 93250183.6

(22) Date of Application: June 21, 1993

(51) Int. Cl.⁵: G07B 17/04

(30) Priority: June 26, 1992 DE 4221270

(43) Date of publication of the application:
December 29, 1993, Patent Gazette 93/52

(84) Designated contracting states:
CH DE FR GB IT LI

(71) Applicant: FRANCOTYP-POSTALIA GmbH
Emmentaler Strasse 132-150
D-13409 Berlin (DE)

(72) Inventor: Heinrich, Klaus
Fontanestrasse 35
W-6453 Seligenstadt (DE)

Inventor: Epping, Thomas
Langgässer Weg 10
W-6103 Griesheim (DE)
Inventor: Günther, Stephan
Forstweg 63a
W-1000 Berlin 28 (DE)
Inventor: Thiel, Wolfgang, Dr.
Bohnsackersteig 8
W-1000 Berlin 27 (DE)
Inventor: Kubatzki, Ralf
Prenzlauer,
Promenade 23
W-1120 Berlin (DE)

(54) A Process and an Arrangement for Rapid Generation of a Security Mark.

(57) A process for rapid generation of a security mark with provision of values by a controller (6) of the franking machine before a print request, with a) Generation of a combination number (KOZ1), b) Encryption of the combination number (KOZ1) into a cryptonumber (KRZ1), and c) Conversion of the cryptonumber (KRZ1) into at least one marking symbol series (MSR1) using a set (SSY1) of symbols. An unmistakable machine-readable and manually evaluable mark is combined column by column during the printing of the entire franking machine image with the other variable data already embedded in the frame data. For this, the symbols are selected from the standpoint of good differentiability and the associated capability of linguistically expressing the image content of the symbol.

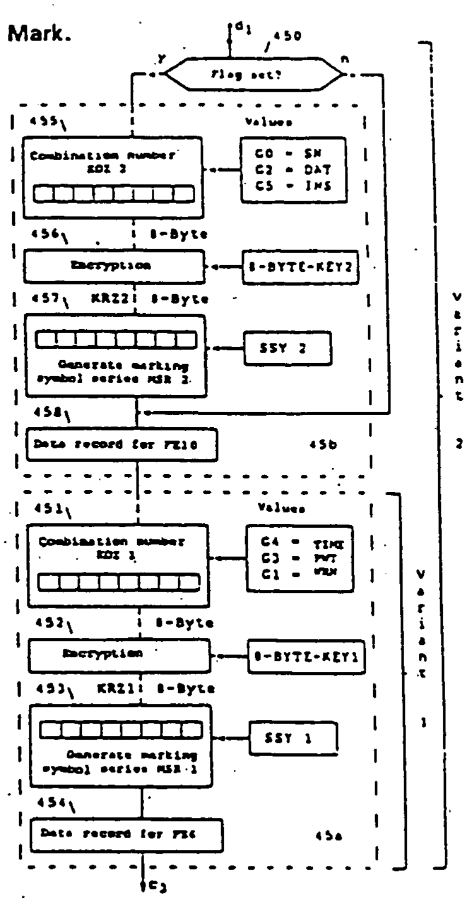


Fig. 10 Step 45: Generation of new coded "Type 2" window data for a marking image

EP 0 576 113 A2

The invention relates to a process for rapid generation of a security mark of the type set forth in the independent element of Claim 1 and an appurtenant apparatus for performing said process. The process comprises steps to prepare variable data immediately prior to printing, in particular to generate marking data and to combine constant and variable data to control the printing in the postage meter in an optimal manner with respect to time for column-by-column printing of a postage meter print image. Among other things, the process permits a rapid change in the text component of an advertisement by means of the editable data stored in the postage meter, without interrupting printing.

Specifically, the invention relates to postage meters which produce a completely electronically generated image for applying postage to pieces of mail, including printing an advertisement and a marking. The postage meter is equipped with at least one input means, one output means, one input/output control module, one memory for at least advertisement, one controller, and one printer module.

As a rule, a postage meter generates a printed image in a right-justified form which is agreed upon with the post office, parallel to the upper edge of the piece of mail beginning with the contents postage in the postage stamp, the date in the date stamp, and stamp impressions for an advertisement and, in some cases, the mail class or type in the optional printing stamp. The postage value, the date, and the mail class/type represent the variable information which needs to be entered for the respective piece of mail.

The postage value is a prepaid mailing fee (postage) paid by the sender, which is subtracted from a reloadable credit register and is used for preparing the article for mailing.

The date is the current date or a future date, located in a mail stamp. While the current date is automatically prepared by a clock/date module, in the case of manual predating, the desired future date must be set. Predating is useful in cases where the mail must be readied and stamped in a very timely manner but must be sent out at a specific date. The variable data for the date can be imbedded in the date stamp as a general rule, as with printing of the postage value.

The approved advertisements may contain messages of various types, in particular the address, company logo, post office box, and/or some other desired message. From the point of view of the postal authorities, the advertisement represents additional information, which is subject to post office approval.

It is sufficiently well known that an advertisement may be placed on a securely positioned slug, which can be changed by the user. A machine of this type produces an unmistakable fingerprint. It would be senseless to copy this impression using a modern color copier since even if the postage value and additionally the serial number are counterfeited, the unique fingerprint permits one to identify the machine which is being imitated for counterfeit purposes. In a solution described in DE 38 40 041, the imprinted portions of the image which do not change continually are placed on a printing drum, and the variable portions of the image (characteristic data) are generated by electronic means and are printed using a thermoprinter. The characteristic data are assigned to a specific location on the image through purely mechanical means.

It is known that the difficulty of changing advertising messages without reducing the security and reliability of the postage fee, can be surmounted through purely mechanical means by changing an individual printing slug containing a text portion of the advertisement. However, this is time consuming and would lead to an interruption in printing. However, such a change could be accomplished much more rapidly with fully electronic impactless printing methods.

A postage meter message printing system in which the characters which are to be printed comprise a postage value and a message communicated from a remote source is already disclosed in DE 37 12 100. This electric postage meter is equipped with a postage meter accounting circuit, which generates the postage amount, with a telephone or communications connection, with a message input device, with a communications control system, with a printer, and with a memory. The memory supplies the message. Thus, the characters which are to be printed are printed based at least in part on the data stored in the memory.

The advertisement may be changed selectively by the data center using the communications control system located between the communications connection and the memory, if the message input device is used to send a request to the data center via the communications connection, if the verification of the coded request in the data center yields a positive result, and if the message communicated from the data center has been verified in the postage meter. It has also been proposed that a third party be given the possibility of inserting an advertisement in his own mail by renting the space for the postage meter message. In this case the third-party message is communicated to the data center. The data center must send the third-party message to the postage meter. The data center also controls the use of the message in the postage meter. However, in such a case measures are always required to ensure that the connection is maintained, that the message is authorized, and that the data are correct.

A message cannot be changed without being verified in the data center. Since the memory is connected directly to the printer and since it stores the printing data for the message, the user cannot check which printing data are currently stored in the memory. In addition, the utilization of the message stored in this way by the user in the postage meter cannot be controlled in any desired manner.

To selectively change the advertising message stored in the memory, either the pixel data for the advertising message must be communicated in completely new form from a central office, whereby the number of imprints is monitored, or the advertising message which is to be printed is manually input in a memory via a keyboard, so that the postage, date information, and text lines can be printed one after another.

A disadvantage of this procedure is that when there is a change in the manual input information, the previous old input information is lost. Only advertising messages stored in the central office can be reaccessed. Another disadvantage is that the change cannot be made without interrupting processing. The combined printing is complicated from an electronic and mechanical standpoint. The use of printing drums or wire matrix printers also produces undesirable noise.

The internal construction of the printing drum, which already has setting wheels for the postage and the date, would also be so complicated that it has already been proposed to locate a separate second printer in the advertisement printing drum. US 3,869,986 discloses the use of a second inkjet printer for printing the variable data.

US 4,580,144 discloses an electronic postage meter apparatus having two thermal printing devices, whereby the first prints the fixed portion of the printed image (postal authority characters/marks and image borders) and the second prints the variable portion of the image (postage and date). This division and separate handling of the variable and constant data permits the printing speed to be increased. However, given the lack of a fingerprint, a security mark *per se* is not present. Thus, an additional mark would also have to be printed.

In a postage meter disclosed in US 4,746,234, fixed and variable information are stored in memory means (ROM, RAM) so that when a letter actuates a microswitch on the transport path ahead of the printing position, said data is read out by means of a microprocessor and so that a print control signal is generated. Both the fixed and variable information are then combined electronically into a printed image and can be printed on an envelope which is to be stamped using thermal printing means. When there are a large number of variable window printing image data which need to be integrated, there is a corresponding delay in the generation of the print control signal. The maximum printing speed which can be achieved when the postage data remain the same is limited specifically by the time required to generate the print control signal. Additional material expense is necessary or a reduction in the printing speed must be accepted if a cryptonumber is to be calculated from the data in order to generate a marking before use as a security imprint. In those cases, poor customer acceptance could be anticipated for such a machine (high price and/or too slow).

EP 294 397 discloses an automatic communication system utilizing user (chip) cards. The user cards are equipped with a microprocessor and a data output unit. The postage meter has a terminal for the user cards with a value processing section, means for programming card microprocessors, and a processing section microprocessor for executing a programmable handling procedure, whereby a graphic is loaded from the card memory via the terminal into the printer memory. The possibility of changing the graphic, however, exists only *in toto*, in other words by reloading an externally modified postage symbol including postage stamp by means of user cards. Here, new advertisements do not increase security *per se*. Such a card can easily be produced electronically, on the other hand, a modified advertisement would suggest a manipulation. However, the postal authorities have not yet provided for evaluating such cases.

DE 38 23 719, on the other hand, discloses a security system for use with a symbol/character imprint authorization device. A memory for the data from the graphic change which are to be loaded and for the data for the respective date is assigned to a processor in the postage meter. If the user requests a monetary change, the processor in the postage meter accesses an external switched telephone device via a connection device (modem), which selects a character/symbol pattern which is to be printed. The printed character/symbol pattern is to be used for verifying the security of the postage meter authorization. Here, however, the entire printed image containing the specific symbol/character pattern must be evaluated by the post office, something which can only be done at great expense.

On the other hand, it has already been proposed that certain hidden or encrypted symbols/characters, bar code, be used for the postage machine imprint, applied with a plurality of print heads as visible or invisible markings on the mail article, in order to be able to identify counterfeits.

Thus, in US 4,775,246 an alphanumeric number,

in US 4,649,266 a single alphanumeric digit in a number, are additionally printed in the postage stamp, whereby subjective errors can occur when such digits or numbers are checked by postal officials. In US 4,934,846 (Alcatel), on the other hand, a machine-readable bar code is printed in a special area next to the postage amount stamp, but the disadvantage of this is that it reduces the available printing area for the postage stamp and/or the advertisement.

The application of such a bar code by means of a separate printer is disclosed in US 4,660,221 and US 4,829,568. In the latter US patent, a character/symbol containing offset elements is also printed. This offset contains the relevant security information. Evaluation is performed, for example, in US 4,641,346, by reading such a symbol/character in a column-by-column fashion and comparing it in a column-by-column fashion with stored character/symbols in order to recover the security information. Evaluation is similarly complicated and can only be achieved if the post office uses expensive equipment.

Since a relatively large amount of space is required to represent the bar code, a two-dimensional bar code has been proposed. However, the disadvantage is that the bar code can only be checked by machines and not also manually. A security system disclosed in US 4,949,381 utilizes imprints in the form of bitmaps on a dedicated marking field beneath the postage meter stamp imprint. Even though the bitmaps are packed in a particularly dense manner, the size which is still required for the marking field decreases the height of the stamp image by the height of the marking field. As a consequence, too much of the area required for the advertisement is lost. A further disadvantage is the high-resolution detection device required to evaluate the marking.

A different security system utilizes imprints in the form of a diagram (US 5,075,862) inside the postage meter stamp imprint. If printing elements fail, the dots in the printed image are missing, which can lead to the detection of an alleged counterfeit. Thus, such markings in the form of a diagram within the postage meter stamp imprint are not particularly reliable. Even if the imprint is flawless, machine evaluation is more difficult since the entire print image must always be evaluated.

DE 40 03 006 A1 discloses a process for marking mail to permit identification of postage meters, whereby a multiple-digit cryptonumber including the date, the machine parameters, the value of the postage, and the advertisement is generated and is temporarily stored on an individual basis. A printer controller which makes the settings for the printer means also inserts the cryptonumber in the print pattern during printing. Thus, a counterfeit or any imitation of the postage meter stamp by a nonpaying postage imprint can be detected by means of the cryptonumber. Even in the case in which a number of users utilize a single postage meter, the user who has manipulated the postage can easily be determined. However, this patent does not relate to a fully electronically generated print image for an impactless printer.

DE 40 34 292 already proposes for security reasons for a fully electronically generated print image to only store a constant portion of the postage meter image in the postage meter and to send the remaining appurtenant variable portion from the data center to the postage meter to put together the final printed image.

In this approach, a communication between a terminal containing a metering module and a central office is necessary to combine the printing data each time postage is applied to mail. This delays printing, which also makes this approach unsuitable for applying postage to large amounts of mail. In this approach the fully electronically generated advertisement is also among the constant data in the stamp image, as is the frame layout of the postage amount and date stamp together with the location data and, in some cases, the postal code. However, the advertisement cannot be partially modified in each postage meter.

The above approaches are either too elaborate to achieve a high printing rate, or they have a plurality of printers or they are not suitable for efficiently combining constant and variable data to generate a print control signal for a single printer.

On the one hand, all previous approaches where editing was possible do not permit the rapid interruption-free change of the fixed portion of an advertisement at a high printing speed of up to approximately 6,000 letters per hour. On the other hand, each message previously only contained the information approved by the post office or, if a single message was to be selected from a number of messages, a selection of approved information. Any deviation either leads to the identification of a counterfeit or to reduced protection against counterfeit if such a deviation is permitted.

The objective therefore was to overcome the disadvantages of the prior art in order to invent a process and apparatus for rapidly generating a security mark. What needed to be developed was a machine-readable as well as manually readable and decodable form of the identification mark, which together with the postage imprint could be visibly applied to the piece of mail or the postage strip, as well as a solution for combining constant and rapidly changing editable data for postage meters and for their print control for column-by-column printing of a postage printed image.

An additional function to be provided is changing the advertisement by means of the controls on the novel printing meter without the need to obtain a prior permission through a central data office or the post office and without thereby adversely affecting the security and reliability of postage accounting and without an article of mail stamped in this manner being sorted out as a counterfeit by the post office. At the same time, however, protection against counterfeit should be increased.

In addition, the post office must be able to easily differentiate between postage meter imprints which have been manipulated for counterfeit purposes and those which have not been manipulated but have an editable advertisement text. In addition, it must be possible to identify the machine which was imitated by the manipulator or which itself was manipulated and to detect the machine which was operated by the user beyond the inspection date and/or this should be able to be determined from the imprint.

This objective is accomplished by the characteristic elements of Claim 1.

Using as a point of departure the desire to get by with one microprocessor and one printing module in a postage meter, it is proposed for the printing of a security mark, that a fully electronically generated printing image embed the variable data in the marking in one or more windows within a fixed frame defined by the postage meter imprint image. In order to efficiently generate the marking data, after completion of all inputs at least one combination number is generated from predetermined values and said combination number is encoded into a cryptonumber in accordance with an encoding algorithm, and this cryptonumber is then converted into a marking. During this process, at least one number assigned to the higher digits in the combination number is a monotonically variable value. Thus, the marking changes with each imprint, which makes a piece of mail stamped in this way unique.

This marking is preferably printed in the form of a bar code and/or a series of symbols in a field in the postage meter image simultaneous to this image by the single printer module. The symbolic quality permits visual evaluation by a trained inspector, who will evaluate the shape and content of the symbols, in addition to machine inspection in the post office. The shape of the symbols, which have orthogonal edges, makes machine reading particularly fast and easy through integral measurement of the print density. Compared to bar codes, the symbol row achieves higher information density, so that space can be saved in the postage meter print image or so that more information can be coded in printed form by means of the graphic symbols.

An important reason why the printing speed is not reduced by the time required to generate the marking data, but rather can even be increased, involves the utilization of a time reserve during printing by the controller microprocessor, which performs the column-by-column embedding of window data.

The process for the rapid generation of a security mark for postage meters which accepts coded marking data prior to a print request after editing window data and/or frame data in the postage meter stamp image from the same controller microprocessor which handles accounting and process control generates a print control signal following a print request, and in this signal the marking data which have been converted to binary pixel data are inserted during column-by-column printing during the course of a special print routine column-by-column in the currently printed column at a predetermined location.

The invention is based on the assumption that after the unit has been turned on the postage is automatically set in the postage amount imprint to a value equal to the last input before the postage meter was turned off, that the date is set in the date stamp corresponding to the current date, and that for the imprint the variable data are electronically embedded in the fixed data for the frame and for all appurtenant data which remain unchanged. Said variable data for the window contents are referred to below in short as window data and all fixed data for the value stamp, the date stamp, and the advertisement stamp are referred to as frame data. The frame data may be fetched from a first memory area of a nonvolatile working memory. The window data may be fetched from a second memory area for the purpose of composing a full representation of a postage image. In the invention, the data from both memory areas are combined into a pixel print image prior to printing in accordance with a freely configurable assignment scheme, and during printing they are completed to form a column in the complete postage meter print image. The variable data which are embedded in the printed column during printing comprise, at the least, the marking data. The time required for advance combination of the entire pixel image with the remaining data is reduced accordingly.

The advance combination occurs in a manner similar to that used with the date in the postage stamp and with the postage amount itself in the value imprint, whereby the variable information can subsequently be enlarged and modified in the window provided for this purpose. It is further assumed that the embedding of a variable text component in a frame for an advertisement can be performed in the same manner as with the other window data and that for this purpose a window is defined within the overall layout of the advertisement.

In order to save time, only those parts of a graphic representation which actually change are stored in the event of a change.

In order to prepare for a change in the text component of an advertisement for postage meters, the process assumes that agreed-upon advertisement types are loaded by means of a modem or chip card and that an advertisement type can be selected in an essentially known manner. In the invention, optional editing of an advertisement text component stored in the postage meter as well as a combination and the display of an overall layout and optionally a storing of the edited text component is additionally performed prior to printout.

The apparatus for changing the advertisement text component for the postage meter possesses in one variant of the invention a pixel memory and a nonvolatile working memory with separate memory areas. In addition to the first memory area for the corresponding frame data, whereby the fixed data are assigned to the various advertisement types for a selection, the second memory area is provided for window data, which may include a number of assignable advertisement text components. The nonvolatile memory is connected to a device which can automatically change the data in this second memory area in a predetermined manner.

Specifically, a message is conveyed in an agreed-upon window of a field for an advertisement in the postage meter imprint, whereby the message directly addresses the audience in plain language. The message may be a text component which has not been agreed upon with the post office and which contains any desired information, for example information about closure dates during holidays, or about exhibitions, conventions, and/or similar events.

In the past, each change requested by a user had to be authorized by the post office. The invention is therefore based on the consideration that a post office approval procedure for advertising types which can be partially added to by the customer may be agreed upon so that the postage meter offers the function of changing alphanumeric text components within the advertisement. This eliminates the need to obtain new advertisements from the postage meter manufacturer or dealer or its data center, which in turn eliminates the associated elaborate security procedure including the transmission of coded signals for advertisement data via modem.

A text change must not reduce the security against counterfeiting nor, when pieces of mail which contain such an advertisement text change are checked for proper postage must they be automatically sorted out as being allegedly counterfeit. For this reason, an appropriate marking including permanent postage meter pixel image data not changed when the advertisemen text component is edited is therefore generated for a security imprint.

Changing an advertisement text component is advantageously based on already stored edited text components, so that when a stack of mail is being printed, the advertisement can be partially changed or the mail can be adapted to current requirements without interrupting the printing operation.

The invention proposes that hexadecimal frame data, window data, and assignment data be transferred into the respective separate memory areas of an additional nonvolatile working memory and be stored there in order to display the print image prior to printing and so that the advertisement text components can be edited. The data are then transferred to the volatile pixel memory and the window data are allocated according to the assignment data in the frame data. The invention permits this to be done efficiently, so that printing speed is high. Advantageous further embodiments of the invention are indicated in the dependent claims and are described in greater detail below together with the description of the preferred embodiment of the invention based on the figures. The figures show:

Figure 1,
Block diagram of a first variant of the postage meter of the invention,
Figure 2,
Flow chart for generating the printed image based on a second variant of the postage meter of the invention having three pixel memory areas,
Figure 3a,
Representation of a security imprint with a marking field
Figures 3b to 3e,
Further variants of the arrangement of marking fields for the security imprint
Figure 3f,
Representation of a set of symbols for a marking field in an advertisement
Figure 4,
Security imprint evaluation device,
Figure 5,
Flow diagram for a third variant of the postage meter of the invention having two pixel memory areas,
Figure 6,
Flow chart of a fourth variant of the postage meter of the invention having one pixel memory area,
Figure 7,
Postage image with associated printing columns,
Figure 8,

Representation of the window characteristic values relative to a pixel memory image stored separately from said image,

Figure 9a,

Decoding of the control code, decompression and loading of the permanent frame data and generation and storage of the window characteristic data,

Figure 9b,

Embedding of decompressed current window data of type 1 into the decompressed frame data after the postage meter has been started or after editing of the frame data,

Figure 9c,

Embedding of decompressed variable window data of type 1 into the decompressed frame data after editing of said type-1 window data,

Figure 10,

Generation of new coded type-2 window data for a marking image,

Figure 11,

Decoding of control code and conversion into decompressed binary window data of type 2,

Figure 12,

Printing routine for combining data from pixel memory areas I and II,

Figure 13,

Printing routine for combining data taken from pixel memory area I and working memory areas.

Figure 1 shows a schematic diagram of the postage meter of the invention having a printer module 1 for a fully electronically generated postage image, which contains an advertisement and/or a marking for a security imprint, input means 2 having at least one control, and having a display unit 3, both of which are connected via an input/output control module 4, a nonvolatile memory 5 for at least the constant components of the postage image as well as with a control unit 6. A character memory 9 supplies the necessary printing data for the volatile working memory 7. The control device 6 has a microprocessor μP, which is connected to the input/output control module 4, to the character memory 9, to the volatile working memory 7, and to the nonvolatile working memory 5, with a cost center memory 10, with a program memory 11, with a transport, respectively feed device, optionally with a strip release 12, an encoder (coding disk) 13, as well as with a clock time module 8.

All alphanumeric characters or symbols are stored in character memory 9 pixel by pixel as binary data. Data for alphanumeric characters or symbols are stored in nonvolatile working memory 5 compressed in the form of a hexadecimal number. Corresponding to the position message supplied by encoder 13 relating to the feed of the mail or the paper strip in relationship to the printer module 1, the compressed data are read out o the working memory 5 and are converted into a printed image containing pixel data with the aid of the character memory 9. The printed image is stored in such a decompressed form in the volatile working memory 7. In order to characterize the invention, the following working memories 7a, 7b, and pixel memory 7c are used, even though physically they preferably are the same single memory module.

Working memory 7b and pixel memory 7c are connected to printer module 1 via a print register (DR) 15 and a printer controller 14 possessing an output logic. The pixel memory 7c is switched on the output side to a first input of the printer controller 14 on whose additional control inputs output signals from the microprocessor control device 6 are present. The apparatus to generate a security imprint for postage meters rapidly has in the nonvolatile working memory 5 a first memory area A (for, among other things, the data for the constant parts of the postage image, among them the advertising frame, whereby an assigned index i identifies the respective frame), and a pixel memory area I in the volatile pixel memory 7c. In order to change the window data rapidly, in particular to make a rapid change in an advertising text component, a second working memory B is present in the nonvolatile working memory 5 and a pixel memory area II in pixel memory 7c for the selected decompressed data for the variable components of the postage image. A stored advertisement text component is also identified by an assigned name or subscript j.

The data for a first assignment of the names of the advertisement text components to the names of the advertisement frame are present in a third memory area C of the nonvolatile working memory 5. This assignment permits the data blocks in memory areas A and B to be addressed and to be automatically called accordingly.

Memory areas A through T in nonvolatile working memory 5 can contain a plurality of submemory areas in which the respective data are present in data blocks. The submemory areas A_i are for i = 1 to m frame or permanent data, B_j for j = 1 to n window data, and B_k for k = 1 to p window data, whereby various assignments between the submemory areas in the various memory areas can be selected and/or stored as defaults.

The number chains (strings) for generating the input data using a keyboard 2 or also which are input by means of an electronic scale connected to the input/output device 4, which calculates the postage - not shown in Figure 1 - are automatically stored in memory area T of the nonvolatile working memory 5.

This ensures that the last input values remain intact when the postage meter is turned off, so that when it is turned on again, the postage is automatically set in the postage amount imprint corresponding to the last input before the postage meter was turned off, and so that the date in the date stamp is set to correspond to the current date. In addition, data blocks in the submemory areas, for example A_i, B_j, C, are also preserved. Control code and run-length-coded frame or window data are alternately contained in each data block of a data memory area A_i, B_j, or B_k.

The corresponding temporary assignment of window data to frame data is made after power-on according to the current or the preset future date by the control device 6, which possesses a microprocessor. Prior to printing, the respective selected frame data for the advertisement stamp, for the post office stamp, and for the postage stamp are written to registers 100, 110, 120..., of a volatile working memory 7a from the nonvolatile working memory 5, whereby during the transfer control coding is decoded and is stored in a separate memory area of working memory 7b. Similarly, the respective selected window data are loaded into registers 200, 210, 220,.... The registers are preferably formed from submemory areas in the memory area of working memory 7a. In a different variant, the aforesaid registers are a component of the microprocessor controller 6.

Through decompression, the run-length-coded hexadecimal data are converted to corresponding binary pixel data. The decompressed binary pixel data, which remain unchanged for a relatively long time in a first pixel memory area I and the binary pixel data which frequently change are placed in a second pixel memory area II. Figure 1 shows a schematic diagram for such a first variant of the design of the invention.

New frame and/or window data can be selected after the insertion and storage of binary pixel data in the first pixel memory area I and the selection of editable window data with subsequent decompression as well as their storage as binary pixel data in the second pixel memory area II, as long as a print request is not present.

For example, if a plurality of identical letters must be provided with the same date and must be ready for mailing using the same postage, whereby in the same time in which input occurs the advertisement text component which is loaded into a second pixel memory area II can be changed, it is particularly advantageous if the security markings are placed in such a window in the postage value stamp or in the date stamp or between both stamps. The reloading of pixel memory area II with the decompressed window data which correspond to the selected data which are stored in compressed form in the memory areas of working memory 5 takes place prior to printing. Combination with the remaining binary pixel data stored in memory area I preferably takes place after a printing request is present during a printing routine. Thus, printing does not need to be interrupted to select the advertisement text component data. The invention therefore permits rapid, interruption-free changing of the advertisement text component and the marking up to a speed of, for example, approximately 6,000 letters per hour, based on 16-bit processor technology.

The number of printed letters having the advertisement setting referred to above is recorded in the postage meter for later evaluation.

In a further embodiment of the first variant, a plurality of separate pixel memory areas for frame data and for window pixel data is prepared in a manner not shown in the figures. The window data include, among others, the current postage value and postage meter-specific data (serial number) in the postage stamp, the date, or additional suitable data (absolute time for item count) in the postal stamp, editable advertisement text component data in the advertisement stamp. The combining of frame and window pixel data takes place once again - as in the first variant - during the execution of a specific print routine.

Figure 2 shows a second variant of the design of the invention. Additional marking data, which are generated specifically for this purpose, encoded, and placed in a sixth memory F, are used for identification. The apparatus for changing the advertisement text component for postage meters now has three separate memory areas in the pixel memory. In addition to the first memory area I for the data relating to the advertisement type (frame), the postage, and the current date, and the second memory area II for a plurality of assignable advertisement text component data, a third memory area III is provided for the marking data. The device which is able to change the data in these first, second, and third memory areas is the same microprocessor of the control device 6 which also executes the accounting and printing routines. The data from the three memory areas are combined during printing according to a previously determined (with certain limits freely selectable) assignment to form an overall layout of an advertisement.

In particular, in a modification of the design shown in DE 40 03 006 A1, mail can be identified based on a cryptonumber-generated marking to permit easy identification of postage meters when the multiple-digit cryptonumber is not generated and temporarily stored utilizing the data values for the entire advertisement, which are stored as a hexadecimal number, but rather using only selected data values for

the advertisement frame and additional data, such as the machine parameter of the value setting and the date.

It is assumed that not only digital or numerical values, such as the number of the advertisement being used, but also data values from the graphic information are used to generate the encrypted information. In contrast to DE PS 40 03 006, any given area from the advertisement to which separate data are assigned in a data block can be used to generate the cryptonumber. For this purpose, individual data are selected from this data block. Here it is advantageous if the column end for each new column to be printed is identified as a control code which follows the run-length-coded hexadecimal data. Run-length-coded hexadecimal data located in the first position in the data block may preferably be used.

In a further embodiment of the design of the invention, the appropriate data for the column-by-column regional graphic information are selected from the data block by means of a physical parameter which is present and/or generated in the machine, specifically by the current date, in order to locate at least a number of data (hexadecimal numbers).

In addition, a plurality of data blocks can be assigned to each advertisement number, whereby each data block contains those data relating to a partial area of the advertisement. In this way, the data block containing the associated data for the column-by-column regional graphic information is selected by means of a physical value which is present and/or generated in the machine, in order to fetch at least a number of data (hexadecimal numbers).

Preferably, those run-length-coded hexadecimal data which correspond to a predetermined print column are combined and encrypted in a specific manner, which is described in conjunction with Figure 10 - together with at least some of the data from the machine parameters (serial number, monotonically variable values, time data, inspection data, for example the number of imprints at the last inspection) and the postage to form a number.

When new coded window data are generated before being stored in a third memory area III, the DES algorithm (Data Encryption Standard) can be used, for example, for a high security standard and additionally, conversion to a special graphic character set can be used.

However, other encryption methods and methods for converting the cryptonumber into a marking or identification are equally suitable.

A bar code can take up a significant amount of space in the postage meter printed image, depending on the amount of information coded in it. Correspondingly, it forces one to increase the size of the postage meter imprint, or it is not possible to reproduce all the information in the bar code imprint.

In the invention, an especially compact imprint consisting of special graphic symbols is utilized.

A mark consisting of, for example, the symbols which are to be printed, may be printed in front of, behind, under, and/or above a field within the actual postage stamp imprint. In the invention this mark can be read manually or it is also machine-readable.

A letter envelope 17 transported under the printer module 1 is printed with a postage meter stamp image. The marking field is located here in a manner which is advantageous for the evaluation in a line below the fields for the value stamp, for the date stamp, for the advertisement, and, in some cases, in the field for the optional added text in the postage meter stamp image.

It can be seen from a representation of a first example of the security imprint shown in Figure 3a that there is good readability for manual evaluation as well as machine readability with high recognition reliability.

The marking field is located in a window FE6 located within the postage meter printed image under the date stamp. The amount stamp, which contains the postage in a first window FE1, the machine serial number in second and third windows FE2 and FE3, possesses, in some cases, a reference field in a window FE7 and, in some cases, information on the number of the advertisement in a window FE9. The reference field is used for presynchronization for reading the graphical character sequence and to obtain a reference value for the light/dark threshold used in machine evaluation. A presynchronization for reading the graphical character sequence is also obtained by and/or in combination with the frame, in particular of the postage amount symbol or the amount stamp.

The fourth window FE4 in the date stamp contains the current date or, in special cases, an entry for the predated date. Beneath this is an eighth window FE8 for a compressed accurate time statement, especially for high-performance postage meters in tenths of a second. This ensures that no imprint is identical to any other imprint, which would make counterfeiting by copying of the imprint using a copying machine senseless.

A fifth window FE5 is provided in the advertisement for an editable advertisement text component.

Figure 3b represents a security imprint having a marking field in the columns between the amount stamp and the date stamp, whereby the vertical portion of the frame of the amount stamp which occurs ahead of the amount stamp is used for presynchronization and, in some cases, as a reference field. This eliminates

the need for a separate window FE7. In this variant, the marking data, whose row of symbols is arranged vertically, can be read in a very brief time, almost simultaneously.

Compared to the window shown in Figure 3a, it is also possible to eliminate additional windows for the overt unencrypted imprint. This increases the printing speed, since fewer windows need to be embedded in the frame data prior to printing and thus since the generation of marking data can begin sooner. In order to achieve simple copying protection, all that is needed is the encrypted imprint by means of marking symbols without using an overt unencrypted imprint of the absolute time in a window FE8. In marking field FE6, the marking data, which are generated based on at least the postage amount and a time count are sufficient unto themselves - as will be explained below using Figure 10.

In a third example of the security imprint shown in Figure 3c, an additional figure not shown in the variant in Figure 3b is located in the postage stamp under window FE1 for the postage amount. Additional information can be communicated here, for example by means of the number of the selected advertisement, nonencrypted but in a machine-readable form.

In Figure 3d in a fourth example of the security imprint, two additional marking fields are located in the postage stamp under and above window FE1 for the postage amount.

In Figure 3e in a fifth example of the security imprint, two additional marking fields are located in the postage stamp beneath and above window FE1 for the postage amount. In this figure, the marking field which is located in the postage stamp above window FE1 for the postage amount has a bar code. Additional information, for example relating to the number of the selected advertisement, can be communicated here in nonencrypted but machine-readable form.

With a symbol set having a relatively small number of available symbols, more symbols must be printed for the same information. In such a case, a symbol row can be printed on two lines or in the form of a combination of the variants shown in Figures 3a to 3e.

The marking form may be agreed upon freely with each postal authority. Any general change in the marking image or in the arrangement of the marking field can be accomplished without difficulty given the electronic printing principle.

The apparatus for the rapid generation of a security imprint for postage meters permits a fully electronically generated postage image formed by the microprocessor-controlled printing process from permanent data and current data. For this purpose, a third memory area C is provided in the nonvolatile working area. This area contains data for an initial assignment of the names of the variable parts to the names of the constant parts. By contrast, the data for the constant parts of the postage image which relate at least to the frame of an advertisement are stored in a first memory area A_i, whereby an assigned name identifies the advertisement frame, and the data for the variable part of the postage image are stored in a second memory area B_j or, for the marking data, in memory area B_k, whereby an assigned name identifies the variable part.

A change or replacement of the symbol set - shown in Figure 3f - can be undertaken at specified intervals, for example regularly upon each inspection of the postage meter in order to increase the security against counterfeit.

Figure 3f shows a representation of a set of symbols for a marking field, whereby the symbols are formed in a suitable manner so that both machine reading and visual evaluation by trained personnel in the post office are possible.

In order to increase counterfeit security, a set of symbols is used which is not contained in the standard character set found on conventional printing devices.

The invention saves space by utilizing a higher information density compared with a bar code in the symbol imprint. Differentiating between 10 print densities is sufficient to achieve a length used to represent the information which is smaller by a factor of approximately three when compared, for example, with the ZIP CODE. This results in 10 symbols, whereby the density in each symbol differs by 10%. With a reduction to five symbols, the density may differ by 20%; however, it is necessary to increase significantly the number of symbols which must be printed if the same information as is shown in the symbol set in Figure 3f is to be reproduced. A set containing a larger number of symbols is also conceivable. The row or rows of symbols is then shortened correspondingly, however the recognition reliability is also reduced correspondingly, so that suitable evaluation equipment in the area of digital image processing are necessary, for example equipment with edge recognition. By consistently using orthogonal edges and avoiding any round surfaces, sufficiently high recognition reliability can be achieved with simple digital image processing algorithms. Such recognition systems use, for example, standard commercially available CCD line cameras and personal-computer-supported image processing programs.

A further advantage compared to a bar code consists in the good readability of the individual symbols located in a row in the marking field due to the

symbolic nature of the image content, and the possibility of verbally describing the image contents for a manual evaluation.

In the preferred variant, the marking field is located at least beneath or in a field in the postage meter stamp image, and a line of such symbols is printed beneath the postage stamp imprint simultaneous to this imprint. The character memory 9 converts a cryptonumber into a symbols-containing code. Specifically, a list which assigns graphical symbols to the individual cryptonumbers and which is selected by an additional physical value, advantageously by the postage amount, is used. The encrypted hexadecimal data are decompressed by means of the character memory, in order to print the code marking which is comprised of the symbols which are to be printed. This too is a machine-readable marking.

The machine-based identification of the symbols in the marking code can be accomplished in two variants: a) by means of the integrally measured print density of a symbol or b) by means of edge detection for symbols.

A simple machine evaluation without elaborate pattern detection is made possible by the quantized difference in the density between the symbols. For this purpose, a suitably focused photodetector is located in a reading device.

This simple machine-based evaluation is possible even with envelopes of various colors. In order to compensate for the different measurement values which are actually obtained, whose differences are based on the different printing conditions or paper types, a reference value is derived from the reference field. The reference field is used for the evaluation of the print density. A relatively high degree of insensitivity, also with respect to failed printing elements, for example of a thermal print head 16 in printer module 1 can be achieved in an advantageous manner with this acquired reference value.

The advantage of using a symbol set of the type referred to above is that based on the specifications of the respective national postal authority it is possible to implement machine reading in a simple manner (for example by the integral measurement of the print density of the symbols) and/or to manually identify an authentic postage meter stamp by means of the conceptual contents of the symbols. The lists which are prepared for each serial number or for each user and which are preferably stored in databases at the data center for all postage meters contain data values for each variable which are used to verify the authenticity of a postage imprint. On the one hand, the correlation of the symbols with listed values and, on the other hand, with a set of symbols which is not shown in Figure 3f, the assignment of meaning and the print density can be specified differently for various users.

A corresponding evaluation device 23 - shown in Figure 4 - for manual identification possesses a processor 26 with a suitable program in memory 28, input and output devices 25 and 27. The evaluation device 23 used by the respective postal authority is connected to a data center 21, which is not shown in Figure 4.

A suitable procedure for verifying security imprints comprises the following steps:

a) Visual determination of the serial number and entry via an input device 25,

b) Visual determination of the postage amount and entry via the input device 25,

c) Visual determination of the graphical symbols and input via an input device 25 which has correspondingly marked function keys,

d) Start of an automatic evaluation, in some cases in cooperation with a data center 21, and signaling of the results of the comparison for display of at least part of the values recovered from the marking for manual verification by a post office inspector.

In a first variant of the evaluation, the graphical symbols are entered one after another in the input device 25 manually by a trained inspector or automatically by means of a suitable reading device 24 in order to reconvert the marking printed on the piece of mail (letter) into at least one first cryptonumber KRZ1. Here the controls, specifically the keyboard, of the input device can be marked by the symbols, in order to facilitate manual entry.

In a second step, the overtly printed values which can be found in the postage meter stamp image, specifically G0 for the serial number SN of the postage meter, G1 for the advertisement (frame) number WRN, G2 for the date DAT, and G3 for the postage amount PW, G4 for the nonrecurring time data ZEIT, as well as at least one value G5 INS which is known only to the postage meter manufacturer and/or is known to the data center and is communicated to the postal authority, in order to generate at least one reference cryptonumber VKRZ1.

The verification occurs in a third step by comparing the cryptonumbers KRZ1 with VKRZ1 in the processor 26 of the evaluation device 23, whereby a signal is issued for authorization in the event of equality or nonauthorization if the comparison produces a negative result (inequality).

A second evaluation variant assists in recovering the individual information from the printed marking and comparing it with the information overtly printed on the piece of mail. If the cryptonumbers for the marking

were generated using a symmetric algorithm (for example the DES algorithm), after the first step of the first evaluation variant, the starting number can be regenerated from each cryptonumber. The starting number is a combination number KOZ and it contains the numerical combination of at least two values, whereby one value is represented by the upper places in the combination number KOZ and the other value is represented by the lower places in the KOZ. The part of the combination number which is to be evaluated - for example the postage amount - is separated off and displayed. This eliminates the second and third step used in the first evaluation variant.

All that is needed for performing the evaluation is a device (laptop) equipped with an appropriate program. Here, values which may not be able to be taken from the postage meter stamp image G1 and in some cases G4 and at least one value which is known only to the postage meter manufacturer and/or known to the data center and communicated to the postal authority G5 are encrypted. These are also recovered from the marking by means of decryption and they can then be compared with the values which are stored user-specifically. The lists stored in memory 28 can be updated by means of a link to the data center 21. Additional information on this is provided in conjunction with step 45 - which is represented in Figure 10 - generation of new coded "type-2" window data for a marking image.

In a third evaluation variant the user manually or automatically enters into the evaluation device values G0, G2, G3, and G4 in order to derive a cryptonumber having the same code and encryption algorithm as the user is utilizing in the postage meter. A marking generated from this cryptonumber is displayed and is compared by the user with the marking printed on the piece of mail (letter envelope). The user is assisted by the symbolic nature of the markings defined in the output unit 25 and printed on the piece of mail.

If a less elaborate encryption algorithm (in some cases a simple asymmetric algorithm) is used, a purely mechanical template, which is not shown in the figures, can be produced which, when properly adjusted displays individual symbols. Thus, in addition to the frame number of the advertising frame, which is recognizable to the viewer, the serial number, the date from the postal stamp and the postage amount from the postage stamp must be set. A series of symbols generated from this data is represented and it can be compared with the series of symbols depicted in the marking field. Verification is performed by comparing the markings in order to establish authorization in the event of identity or nonauthorization when the results of the comparison are negative (nonidentity).

The first value G1 is the advertising frame number WRN, which the inspector recognizes from the postage stamp image. This first value is known to the user as well as the postage meter manufacturer and/or data center and is communicated to the postal authority. In one variant, preferably having a data link to the data center, the advertising frame WR_n with associated numbers WRN_n, which belongs to the serial number SN of the respective postage meter, is displayed on a screen of the data output device 27. The comparison with the advertisement frame WR_b used on the letter [b = *Brief* = letter] is performed by the inspector, who enters the WRN_n determined in this manner.

The stored lists communicated by the data center into the memory 28 contain the current assignment of the advertisement frame WRNT to a second value G2 (for example the date DAT), and the assignment of symbol lists to a third value G3 (for example the postage amount PW). In addition, a list of parts SNT of the serial number SN selected by means of the first value G1, specifically by the advertisement (frame) number WRN may be present. A user-specific piece of information, for example the advertisement frame number WRN may be used for a random-sample manual evaluation of the marking, in that decoding lists which contain the appropriate data blocks can be selected on the basis of user-specific information. The value G2 (DAT) is then used to select the byte from the data block which is used to generate the combination number.

In the preferred variant, a monotony test is used to verify the uniqueness of the imprint. The inspector acquires the serial number SN from windows FE2 and FE3 of the imprint and determines the postage meter user. The advertising number can also be used for this purpose, since as a rule this number is assigned to specific cost centers when the same machine is used by various users. Data on the latest test as well as data on the latest inspection are entered in the above lists. Such data are, for example, the number of pieces of mail, if the machine has an absolute piece counting system, or the absolute time data, if the machine has an absolute time counter.

The correctness of the printed postage amount according to postal regulations is verified in the first test step. In this step it is possible to detect after-the-fact manipulations on the postage amount imprint made with fraudulent intent. In the second test step, the monotony of the data, in particular of the data in window FE8 is checked. This can be used to detect copies of a postage imprint. Manipulation for counterfeit purposes therefore is not a promising activity, since these data are also printed in the form of an encrypted series of symbols in at least one marking field.

With absolute time or piece counting, the number indicated in window FE8 must have increased since the last test was performed. Nine digits are shown in window FE8, which permits a time period of approximately 30 years to be registered at a resolution of seconds. The counter would not run out until after this period. These values can be recovered from the marking so that they can be compared with the overtly printed unencrypted values.

In a third optional test step, if there is suspicion of manipulation, the other values can be checked and identified, in particular the serial number SN of the postage meter, in some cases the cost center of the user.

The information, such as the advertising (frame) number WRN, can also be stated in a predefined window FE9. The data belonging to this window are of type 1, in other words they change less frequently than do the type-2 window data, for example the time data in window FE8 and the marking data in FE6.

In a further embodiment, the data in windows FE8 and FE9 are not overtly printed out in unencrypted form. Rather, they are used only for encryption. For this reason, windows FE8 and FE9 shown in Figure 3a are missing from the postage meter imprint images represented in Figures 3b to 3e, which are used to illustrate these embodiments.

In a preferred input embodiment for the test, the temporary variable values which are to be input, for example the advertisement (frame) number WRN, the date DAT, the postage amount PW, the time data ZEIT and the serial number SN, are automatically detected and read in by means of a reading device 24, in each case from the respective field in the postage meter stamp image. Here, the arrangement of the windows in the postage meter imprint must be maintained in a predetermined manner.

Other temporary variable values associated with the respective serial number SN are known only to the postage meter manufacturer and/or data center and are communicated to the postal authority. For example, the defined number of pieces of stamped mail attained at the last inspection serves as the fifth value G5.

All values which are to be entered, with the exception of values G1, G4, and G5, must be able to be taken from the individual windows FE_j of the postage meter stamp image. In this process, value G5 for example provides the key for the encryption, and it is changed in specified time intervals, for example after each inspection of the postage meter. These time intervals are to be measured in such a way that even when modern methods of analysis are used, for example differential cryptoanalysis, it will be certain that the original information cannot be reconstructed from the markings in the marking field in order to create counterfeits of postage meter stamp images.

Value G1 corresponds, for example, to an advertisement (frame) number. The corresponding number chains (strings) for window and/or frame input data are stored in submemory areas T_i, T_j of working memory 5 of the postage meter.

Values G0, G2, and G3 correspond, for example, to the window data stored in memory areas T_j of working memory 5 of the postage meter, whereby value G0 in windows FE2 and FE3, comes from submemory areas T_2 and T_3, value G2 in window FE4 comes from submemory area T_4, and value G3 in window FE1 comes from submemory area T_1.

The following steps are provided for changing advertisement text components for postage meters:

a) Load agreed upon advertisement frames via modem or chip card,

b) Select an advertisement frame,

c) Edit an advertisement frame which is automatically stored prior to the application of postage,

d) Compose and display a complete representation of the stamp image.

If steps a) through c) have already been formed previously, the stored advertisement frame together with the embedded frame text components can be displayed and selected immediately, in order to make a change in the advertisement or the combination of an advertisement with the selected advertisement text component without interrupting printing.

A representation of a number of names of advertisement frames or a plain-language representation for the selection of advertisement frames with an advertisement or standard text component assigned tc them from a pixel memory is further provided in the display unit 3 of the postage meter for the purpose of selecting an advertisement frame. Acknowledging the selection assigns at least one of the variable text components for FE5 to the respective advertisement frame. The controls can be used to edit the selected advertisement text component while the text is displayed in plain language. After an advertisement text component for window FE5 has been added and the overall representation of a stamp image has been composed and displayed, the edited text component is automatically saved prior to application of the postage, whereby a new data block or a new assignment of the edited text component to the advertisement frame is stored in memory areas B_5 or C.

The windows within the overall representation of the advertisement imprint are defined in such a way that only those parts of a graphical representation which were actually changed are stored in the event of

a change. They are stored in a separate submemory area B_5. The data for the contents of the window can be recovered from memory area B so that they can be combined into a full representation of a stamp image. Run-length coding of graphical data is provided in at least one area of the nonvolatile working memory 5, whereby the first code in each line of the coding declares the number of pixels to be printed per column, and control codes, for example for the start of the respective window (type 1 or 2), column end, image end are present in hexadecimal form.

The advertisement frame data and the data for the variable advertisement text components are found as needed in memory areas A_i and B_5, whereby attention is paid to the assignments. Preferably a plurality of data blocks for the advertisement text components are stored in submemory area B_5.

When the postage meter is started, the user enters a cost center, which causes a predetermined advertising frame data block to be selected from memory area A_i. The controller 6 can perform the predetermined combination of the printing image data since the selectable advertisement frames are each assigned to a cost center KST and/or a second assignment of frame type number to cost center is stored in a fourth memory area D.

The names of the advertisement frames stored in a first memory area A are put together from the number KN of the cost center K and a type number TN. The names of the advertisement text components, which are stored in a second submemory area B_5, by means of which the current date at the time of generation and optionally by means of the time of day UZ, and the data for the first assignment are determined in the third memory area C by storing both the name of the advertisement frame which contains the cost center number KN and the type number TN as well as the name of the advertisement text component, which contains a date DAT and, in some cases, a time of day UZ. Depending on which cost center is selected, an assignment is found in memory area D, and a predetermined data block for the advertisement frame is selected from A_i. The associated advertisement text component can be addressed in memory area B_5 using the assignment given in memory area C, and can then automatically be transferred into the memory area 7a intended for this purpose in the volatile working memory 7.

In addition, the representation in display unit 3 needed for selection of an advertisement is only made of the names of those advertisement frames for which assignments which fall within a defined time period exist in a third memory area C. An advertisement frame for which an assignment of a name to an advertisement text component name does not exist can no longer be displayed. The assignment can be based, for example, on the time period within the current year. After the name has been selected and acknowledged, the appropriate advertisement frame is shown in a WYSIWYG-like representation.

The representation of names of advertisement text components takes place in the window provided for this purpose of the field in display unit 3 in which the WYSIWYG-like representation of the advertisement frame is visible.

For a different embodiment of the representation, the WYSIWYG-like representations of the advertisement text components appear directly one after another in an order which is based on the time data in the name in order to make it easier to select advertisement text components.

In a preferred variant, the defined time period is automatically provided by the selectable time data in the data for the assignment of advertisement text components, in particular such time data as, for example, the month and/or year, in relation to the current date automatically supplied by a clock/date module 8.

In a further preferred embodiment of the idea of the invention, an assignment is made in a fifth memory area E which [the assignment] relates to a freely programmable defined time period in which the time data in the data for the assignment of advertisement text components to advertisement frames are related to the current date.

The stored window data for an advertisement text component, a marking field, and, in some cases, for a reference field are located in the submemory areas B_5, B_6, and B_7 of working memory 5. Here it must be noted that in some of the submemory areas identified as B_k of working memory 5 of the postage meter, the window data are read in and/or read out more frequently than in other submemory areas. If the nonvolatile working memory is an EEPROM, a special storage method can be used to be certain to remain within the limited number of memory cycles permitted for this memory. On the other hand, a battery-backed-up RAM can also be used for the nonvolatile working memory 5.

The window data which change less often with time are referred to as type-1 window data. On the other hand, frequently changing window data are referred to as type-2 window data.

Compared to the second embodiment shown in Figure 2, Figure 5 now shows a third embodiment of the invention, whereby the process is based on the existence of two pixel memory areas shown in Figure 1.

Depending on the frequency with which data change, coded binary frame and window data are

stored in two pixel memory areas prior to printing. The type-1 window data which are not continually changing, such as date, serial number of the postage meter, and an advertisement text component which is selected for a number of printings can be decompressed together with the frame data as binary data prior to printing and combined to form a pixel image stored in pixel memory area I. On the other hand, constantly changing window data of type 2 are decompressed and are stored as binary window data in the second pixel memory area II prior to printing. Type-2 window data are the postage amount, which depends on the item of mail and the form and class of the mail, and/or the continually changing marking. After a printing request, and during the course of a printing routine during printing, the binary pixel data for each column in the printed image are taken from pixel memory areas I and II and combined into a print column control signal.

After the start in step 40, an automatic input of the most recent currently stored window and frame data takes place based on the entry of the cost center in step 41. The corresponding display appears in step 42. An advertisement text component, which is assigned to a specific advertisement is automatically specified in the manner described above.

In step 43 the frame data are moved to registers 100, 110, 120,... of the volatile working memory 7a and the control code is detected and stored in volatile working memory 7b. The remaining frame data are decompressed and stored in volatile pixel memory 7c as binary pixel data. Similarly, the window data in registers 200, 210, 220,... of volatile working memory 7a are loaded and control code is detected and stored in volatile working memory 7b, and the remaining window data are then stored in volatile working memory 7c column-by-column after they have been decompressed.

Figure 9a shows in detail the decoding of the control code, decompression and loading of the permanent frame data and the generation and storage of the window characteristic values, and Figure 9b shows the embedding of decompressed current window data of type 1 into the decompressed frame data after the postage meter has been started or after frame data have been edited.

In step 44 either the decompressed frame and window data of type 1 are present stored as binary pixel data in pixel memory area I and can be processed further in step 45, or new frame and/or window data are entered. In the latter case, the procedure branches to step 51.

In step 51 the microprocessor determines whether an input has been made via input means 2 in order to replace window data, for example those used for the postage amount, with new data or to replace or edit window data used, for example, for an advertisement text line. If such an input has occurred, the substeps required for the inputs are performed in step 52, in other words, a different completed data block is selected (advertisement text components) and/or a new data block is generated which contains the data for the individual characters (letters and/or numbers) of the input value.

In step 53 the corresponding data blocks for a display used to check the input data are called and they are prepared for subsequent step 54 for reloading type-1 window data into pixel memory area I.

Figure 9c gives a detailed representation of step 54 for the embedding of decompressed variable window data of type 1 in the decompressed frame data after a new input or after the editing of these type-1 window data. The data from data blocks called in accordance with the input are evaluated in order to detect control code for an "inking change" or an "end of column," which are necessary for embedding the newly inputted window data. Data which do not contain a control code are then decompressed into window pixel data and are embedded column-by-column in pixel memory area I.

If, by contrast, it was determined in step 51 that no window data are to be selected or edited, then the procedure branches to step 55. In step 55, the possibility of changing the permanent advertisement or frame data which are in use leads to step 56 in order to input the currently selected frame data blocks together with the window data blocks. Otherwise, the procedure branches to step 44.

If selected specific values are to be entered anew, a flag is set in step 44 and in the following step 45, which is used to generate data for a new series of marking symbols, this flag is acted upon if here a step 45b is processed in a second variant.

New coded type-2 window data are generated in step 45. Here, marking data are preferably generated for window FE6, whereby the previous steps used to encrypt data for generation of a cryptonumber are included. Forming into a bar code and/or chain of symbols is also provided for in step 45. Figure 10 illustrates the generation of new coded type-2 window data for a marking image based on two variants. In the first variant, a monotonically variable value is processed in step 45a in such a way that the printed series of marking symbols makes each imprint unmistakably unique. In a second variant, still other values are processed in step 45b ahead of step 45a.

The data block for the marking data which is formed in this way is then loaded in area F and/or at least one submemory area B_6 of the nonvolatile working memory 5, overwriting the previously stored data block

for which window characteristic values have already been determined or are predefined and just now are stored in volatile working memory 7b. Submemory area B_{10} is preferably provided for a data block which results in the printing of a second row of marking symbols, as is shown in Figures 3c and 3d. Moreover, double rows of symbols can be printed next to one another in a manner not shown in Figure 3b. Area F is preferably provided for a data block which results in the printing of a bar code, as is shown in Figure 3e.

In step 46 the data from the data block for the marking are transferred into registers of volatile working memory 7a and the control codes "inking change" and "end of column" are detected so that the remaining data in the data block can be decoded and so that the decoded binary type-2 window pixel data can be loaded into pixel memory area II of volatile working memory 7c. Figure 11 provides a detailed representation of the decoding of control code and conversion into decompressed binary type-2 window data. Such type-2 window data are identified in particular using the subscript k, and they relate to the data for window FE6 and in some cases FE10 for marking data and in some cases FE8 for the ZEIT (time) data from the absolute time counting system. The time data in particular represent a monotonically varying value since they increase in a time-dependent manner. Time data which are still BCD-packed are supplied from the clock/date module 8 and then, if necessary, converted into a suitable data block containing ZEIT data with run-length-coded hexadecimal data. Now they can also be saved in memory area B_8 for type-2 FE8 window data and/or in step 46 immediately loaded column-by-column into register 200 of working memory 7a or into the print register 15.

Step 47 waits in a waiting loop for the print request when step 48 contains a print routine as a response to a print request and when the print request has not yet been made. In one embodiment, the waiting loop returns directly to the beginning of step 47, in the manner shown in Figures 5 and 6. In a different embodiment, the waiting loop returns to the beginning of steps 44 or 45 in a manner not shown in Figures 5 or 6.

The printing routine for combining printing column data from pixel memory areas I and II which is performed in step 48 and is shown in detail in Figure 12 takes place during loading of the print register (DR) 15. Immediately after the print register (DR) 15 has been loaded, the printer control (DS) 14 causes the loaded print column to be printed. Then a check is performed in step 50 to determine whether all the columns needed for a postage meter print image have actually been printed. This is done by comparing the running address Z with the stored end address Z_{end}. If the printing routine for a piece of mail has been completed, processing branches to step 57. Otherwise it branches back to step 48, to generate and print the next print column until the printing routine has been completed.

Once the printing routine is completed, step 57 checks whether additional pieces of mail need to be stamped. If they do [sic], then the application of postage is terminated in step 60. Otherwise, the end of printing has not yet been reached and processing branches back to step 51.

Figure 6 shows a fourth embodiment of the design of the invention whereby, deviating from the block diagram in Figure 1, only one pixel area I is used. Decoded binary frame data and type-1 window data are combined and stored in this pixel memory area prior to printing. The steps are identical up to step 46, which is eliminated here in this variant shown in Figure 6, and step 48, which is here replaced by step 49. Up to step 46 the processing sequence is essentially identical.

Figure 13 gives a more detailed representation of the print routines used for combining data taken from pixel memory area I and working memory areas.

The constantly changing type-2 window data are decompressed in step 49 during printing and are combined with the binary pixel data from pixel area I, which are to be printed column-by-column, to form a print column control signal. Type-2 window data are, for example, the postage amount, which depends on the mail article which is to be printed and the mail class/type, and/or the constantly changing marking.

The generation of the print control signal from frame and window data is illustrated using a postage amount image - shown in Figure 7 - and the data in the print control signal assigned to one print column. A letter envelope 17 is transported under the print module 1 of an electronic postage meter at velocity v in the direction indicated by the arrow and the represented postage amount image is raster printed column-by-column beginning in column s_1. The printer module 1 has, for example, a print head 16 containing a row of print elements d1 to d240. The inkjet printing method or a thermal transfer printing method, for example the ETR method (Electroresistive Thermal Transfer Ribbon) can be used for printing.

A column which is to be printed s_i has a print pattern 30 which is to be printed and which consists of inked print dots and non-inked print dots. One inked print dot is printed with one print element. The non-inked print dots are not printed. The first two print dots in print column s_i are inked in order to print the frame 18 of the postage amount image 30. Then fifteen non-inked (i.e. not active) and three inked (i.e. active) print dots are alternatingly printed until a first window FE1 in

which the postage amount is to be inserted has been completed. There then follows an area of fourteen non-inked print dots until the end of the column. This type of run-length coding is implemented in the data block by means of hexadecimal numbers. The amount of storage space required for the data is minimized by the fact that all data are in such a compressed form.

Two hundred fifty-six bits can be generated with hexadecimal datum "QQ." If one subtracts the necessary control code bits from this number, there remain less than 256 bits for controlling the means used to generate the dots.

However, if one additionally uses control code "00" to indicate an inking change, even more than 256 dots can be controlled, whereby however more memory is required in submemory area A_i of working memory 5. The typical embodiments in Figures 9, 11, 12, and 13 are set up for such a high-resolution printer module.

Control codes are provided "00" for inking changes. This means that a following hexadecimal number continues to be evaluated as inked ($f = 1$), which otherwise would not be inked. A reset inking flip-flop ($f = 0$) is used when an inking change is made ($f = 1$) and it then switches upon the next inking change ($f = 0$). Two hundred fifty-six or more dots can be addressed with this principle. Register 15 in the print control 14 is loaded bit-by-bit from the pixel memory (for example for a print column having $N = 240$ dots).

Additional control codes are "FE" for end of column, "FF" for end of image, "F1" for the beginning of the first window FE1, etc.

In the following example used to illustrate Figure 7, for a print column having more than 240 dots which must be controlled, less space is required in ROM, since the control codes are set up in an advantageous manner. For hexadecimal data "01", "02",... , "QQ", ... "FO" up to 240 dots can be controlled ("FO" $= [F*16^1] + [0*16^0] = [15*16] + [1] = 241$).

Here, control code "00" for inking changes can theoretically be eliminated, since using a single hexadecimal number "FO" an entire print column of 240 dots having the same inking pattern can be completely defined. However, with only an imperceptibly slight increase in memory requirements, it may make sense to use inking changes if there are more than one window in one column.

Using this method, the data block for the printing column s_i of the form shown in the cutout is:

..."2", "0D", "02", "4F", "F1", "68", "FE",..., ...

When the data are moved into register 100 in the µP control 6, "QQ" control codes are detected in the hexadecimal numbers and are evaluated during the course of step 43.

During this evaluation window characteristic values Z_j, T_j, Y_j respectively Z_k, T_k, Y_k are generated and stored together with the values determined for the starting address Z_0, end address Z_{end}, and the total run length R, in other words the number of binary data required per printing column are stored in RAM volatile memory 7b.

A maximum of 13 windows can be called for the 13 control codes "F1" to "FD" and the starting address determined. For example with "F6" for the beginning of a window FE6 of type 2, a starting address Z_6 can be determined and stored as a window characteristic value.

Figure 8 represents the window characteristic values for window FE1 which are related to a pixel memory image and are stored separate from this image. The window has a window column run length $Y_1 = 40$ pixels and a number of columns of approximately 120, which is stored as a window column variable T_1. If the window start address Z_1 is stored as the destination address, then the position of window FE1 in the binary pixel image can be reconstructed at any time.

Binary data converted from registers 100, 200 are read into the volatile pixel RAM memory 7c bit by bit, whereby an address is assigned to each bit. If the hexadecimal number loaded into the register is a detected control code "F2", the window characteristic value Z_j for a starting address of the window of No. $j = 2$ is determined with a total of n windows. In this way, window data can later be reinserted in the frame data at this location, which is indicated by the address. The window column run length Y_j is less than R the total run length of the print column. The new address in the same line but in the next column can be generated by adding R.

Figure 9a shows the decoding of the control code, decompression, and loading of the window frame data, as well as the generation and storing of the window characteristic values. Taking into account the generation of extremely high-resolution imprints, a "inking change" control code was used. For this reason, an inking flip-flop 1 of $f = 0$ must be reset in an initial substep 4310. Let the source address H_i for finding the frame data initially be $H_i = H_i - 1$ and the destination address $Z = Z_0$.

For the type-1 window data, in substep 4311 the window column variable is set $T_j = 0$, for $j = 1$ to n windows and for the type-2 window data, the window column variable $T_k = 0$ for $k = 1$ to p windows. In substep 4312 the source address H_i for the frame data is incremented and an inking change is performed, so that the initial data byte is evaluated for example as inked, which subsequently causes the corresponding print elements to be activated.

The above byte, which is a run-length-coded

hexadecimal number for frame data, is now transferred in substep 4313 from the corresponding area A_i, which is automatically selected by the cost center KST, of nonvolatile memory 5 to register 100 of volatile memory 7a. During this transfer, control codes are detected and a run length variable X is reset to zero.

In substep 4314 a control code "00" for an inking change is detected, which, following a branch back to substep 4312, results in an inking change, in other words the next run-length-coded hexadecimal number causes a deactivation of the printing elements for the given run length. If no detection is made, then substep 4315 determines whether control code "FF" for the end of an image is present. If such a control code is detected, point d in Figures 5 or 6 is thus reached and step 43 has processed to completion.

If, on the other hand, in substep 4315 the control code "FF" for end of image is not detected, then substep 4316 checks whether the control code "FE" for end of column is present. If such a character is detected, the inking flip-flop 1 is reset in substep 4319 and processing branches to substep 4312 in order to then load the byte for the next printing column in substep 4313. If no end of column is present, substep 4317 determines whether a control code for a type-2 window is present. If such a control code is detected, processing branches to substep 4322. However, if it is not detected, substep 4318 checks whether a control code for type-1 windows is present. If this is the case, then point c_1 is reached at which step 43b - shown in Figure 9b - is performed.

If a control code for type-1 window data is not detected in substep 4318, then the called byte contains the run-length-coded frame data, which are decoded in substep 4320 and converted into binary frame pixel data and stored in pixel memory area I of pixel memory 7c at the set address Z. In the following substep, 4321, the column run length variable X is determined based on the number of converted bits and then the destination address for the pixel memory area I is increased by this variable X. At this point, point b has been reached, and processing branches back to substep 4312 to call a new byte.

In substep 4322 if a control code for type-2 window data were present, the executed storage of window characteristic values T_k would be determined. If a window characteristic value - in this case the window column run variable T_k - is still at the initial value of 0, the window starting address Z_k is determined in substep 4323 according to the address Z and it is stored in volatile working memory 7b. If this is not the case, processing branches to substep 4324. Substep 4324 also follows substep 4323. In substep 4324 the window characteristic value of the window column variable T_k is incremented. In the following substep, 4325, the window variable T_k previously stored in volatile working memory 7b is written over with the current value, and point b is reached.

The window characteristic values are loaded in this way for k = 1 to p windows, specifically FE6 in some cases FE10 or FE8. Processing then branches back to substep 4312 in order to load a new byte in substep 4313. The bits converted from the hexadecimal data (dot = 1) are thus moved bit by bit in step 43a - shown in Figure 9a - into pixel memory I of volatile pixel memory 7c and are stored in sequence as binary data.

Figure 9b shows the embedding of decompressed current type-1 window data in the decompressed frame data after the postage meter has been powered on or after the editing of frame data. Assuming that a control code for type-1 windows has been detected in substep 4318, point c_1 and hence the beginning of step 43b is reached.

Substep 4330 determines whether window characteristic values T_j have been stored. If a window characteristic value, in this case the window column run variable T_j, is still at an initial value of zero, then in substep 4331 the window starting address Z_j corresponding to address Z is determined and stored in volatile working memory 7b. If not, processing branches to substep 4332. Substep 4332 follows substep 4331. In substep 4332 the window characteristic value of window column run length Y_j and the window column run length variable W_j are set to an initial value 0 and the window source address U_j is set to the initial value U_{oj} -1 and the second inking flip-flop for windows is set to "Do Not Print Inked."

In the following substep, 4333, the previous window source address U_j is incremented and an inking change is performed, so that any window bytes which are loaded in the following substep 4334 are evaluated as inked, which then during printing causes the printing elements to be activated.

In substep 4334 a byte from submemory area B_i in nonvolatile working memory 5 is loaded into register 200 of volatile working memory 7a and checked thereby for the presence of control codes.

In substep 4335 the window column run length Y_i is incremented by a value equal to the window column run length variable W_j. In substep 4336 a check is performed to determine whether a control code "00" or an inking change is present. If such a code is detected, processing branches back to substep 4333. If it is not detected, substep 4337 checks whether control code "FE" for end of column is present. If this is not the case, then window data are present. In such a case substep 4338 decodes the contents of register 200

with the aid of character memory 9 and it stores the binary window pixel data corresponding to this byte in pixel memory area I of pixel memory 7c.

The window column run length variable W_j is then determined in substep 4339 in order to increment the address Z by an amount equal to the value of variable W_j. The new address for the byte from the data block which is now to be converted is now available, and processing branches back to substep 4333 in which the new source address for a byte from the data block for window FE_j is generated.

If the control code "FE" for an end of column is detected in substep 4337, processing branches to substep 4340, in which the window column variable T_j is incremented and the window column variable T_j and the window column run length Y_j stored [in] volatile working memory 7b are written over with the current value. An inking change is then performed in substep 4341, and point b is reached.

Thus, step 43b has been processed to completion, and new frame data could be converted in step 43a, if the next window is not detected or if point d has been reached.

Figure 9c shows the embedding of decompressed variable type-1 window data in the decompressed frame data after editing of these type-1 window data. As has already been shown, pixel memory data and window characteristic values are already stored prior to the beginning of step 54.

Substep 5440 begins with the determination of that number n' of windows for which the data have been changed and with a determination of the corresponding window starting address Z_j and window column variable T_j for each window FE_j. In addition, a window counting variable q is set to zero.

Substep 5441 determines whether the value of the window counting variable q has already reached the value of the number of window changes n'. In the case of zero changes, in other words n' = 0, the comparison is positive and point d is reached. Otherwise, processing branches to substep 5442, whereby for a first window FE_j whose data were changed, the window starting address Z_j and the window column variable T_j are taken from volatile working memory 7b. In addition, the source address U_j is set to an initial value U_{oj} - 1, which uses the destination address Z_j for addressing pixel memory area I; and a window column counter P_j, and the second inking flip-flop are set back to the initial value of zero.

The following substep, 5443, increments the source address and an inking change is made before substep 5444 is reached. In substep 5444, a byte of the changed data block in the nonvolatile memory is called and is transferred into register 200 of volatile memory 7a with detection for control codes. If control code "00" for an inking change is present, substep 5445 branches back to substep 5443. If not, it branches to substep 5446, to look for control code "FE" for end of column. However, if such a control code is not present, the contents of register 200 can be decoded in the following substep, 5447, using the character memory 9 and can be converted into binary pixel data for the window which is to be changed. These data replace the pixel data previously stored in area I of pixel memory 7c beginning at the location predetermined by window starting address Z_j. The bits which are converted in this process are counted as window run length variable W_j, with which the destination address V_j is incremented in substep 5448. Processing then branches to substep 5443 in order to load the next byte in substep 5444.

If control code "FE" for end of column is detected in substep 5446, however, then processing branches to substep 5449, in which the window column counter P_j is incremented.

Substep 5450 uses window column counter P_j to check whether the window characteristic value for the corresponding window column variable T_j has been reached. If so, all the changed data for a first modified window will have been loaded into pixel memory area I and processing will branch to substep 5453 and from this substep to substep 5441, in order to transfer changed data for a possible second window into pixel memory I. In substep 5453, the window counting variable q is incremented for this purpose, and the following window starting address Z_{j+1} and the following window column variable T_{j+1} are determined.

If not, and if in substep 5450 the window column variable T_j has not yet been reached by the window column counter P_j, processing jumps back to substep 5443 via substeps 5451 and 5452, in order to overwrite an additional window column in the pixel memory area until the old binary window pixel memory data have been completely replaced by the new data. In substep 5451 the destination address for the data in pixel memory area I are incremented for this purpose by an amount equal to the frame total column length R. Thus, the destination address V_j is set to the next column for binary pixel data of the window in the pixel memory area I. Substep 5452 resets the inking flip-flop to zero, so that conversion can begin with pixel data which are evaluated as "inked." If no further new entry is detected in step 44, the generation of new type-2 coded window data for a marking image can be performed in step 45, in particular based on a first embodiment containing step 45a.

Step 45 includes additional substeps - which are shown in Figure 10 - for generating new type-2 coded

window data for a marking image.

While already decompressed binary pixel data are present in pixel memory area I, following step 44 in step 45 the initial data for the data blocks containing the decompressed data for windows FE_j and, in some cases, for the frame data are required so that new coded type-2 window data can be generated for the row of marking symbols. The individual initial data (respectively input data) are stored in memory areas T_w as a BCD-packed number corresponding to the respective values G_w. In addition to the nonvolatile data blocks stored in submemory areas A_i and B_j, the data for a data block for window FE_k of type 2 are compiled in several steps and are stored in a nonvolatile manner in submemory area B_k.

The process for rapidly generating a security imprint comprises, after values have been readied, substep 45a performed by a microprocessor in the controller 6 of the postage meter prior to a print request (step 47). Substep 45a is further broken down into the following substeps:

a) Generation of a combination number KOZ1, whereby a uniformly monotonically variable value G4 is made available to generate initial interrelated positions and at least one additional value G3 describing the item of mail is provided to generate second interrelated positions in the combination number KOZ1,

b) Encryption of combination number KOZ1 to form cryptonumber KRZ1,

c) Conversion of cryptonumber KRZ1 into at least one row of marking symbols MSR1 using a set SSY1 of symbols.

In a first variant 1 a row of marking symbols is generated in step 45a. In a manner corresponding to the invention, based on the set of information and by means of values G0 to G5, which are only partially overtly printed in unencrypted form in the postage meter stamp image, at least part of the values are used in the postage meter to form a single combination of numbers (substep 451) which is encrypted into a single cryptonumber (substep 452), and then converted into a marking which is to be printed on the piece of mail (substep 453). The data block which is to be generated for the marking in window FE6 can then be stored in a concluding substep 454. Point c_3 is then reached. This first embodiment which is performed in substep 45a can then save the time which is otherwise required in the postage meter for generating additional cryptonumbers.

The uniformly monotonically variable value G_w is at least an ascending or descending machine parameter, specifically a time count or its complement during the service life of the postage meter.

It is advantageous if a machine parameter is tim[e] dependent, in particular if it comprises a value G4[a] which characterizes the decreasing battery voltage o[f] the battery-backed-up memory and a second uniforml[y] monotonically descending value G4b or the respective complements of the values G4a and G4b.

In one variant the second uniformly monotonicall[y] decreasing value G4b is the complement of the piece count or it is a uniformly monotonically decreasing time-dependent value.

In one variant the uniformly monotonically decreasing value is a numerical value corresponding to the next inspection date (INS) and a uniformly monotonically decreasing time-dependent value.

In another variant, a uniformly monotonically increasing value comprises the date or the mail piece count determined at the last inspection.

It has already been described in detail how it is advantageous if, to generate third interrelated positions in combination number KOZ1, a portion of the values G0, G1 which characterize the user of the postage meter is made available by the controller 6.

Preferably, in substep 451 the upper 10 places in combination number KOZ1 are readied from memory area T_w for the ZEIT data (value G4) and the lower four places are readied for the postage value (value G3). This results in a combination number having 14 digits, which then needs to be encrypted. When the DES algorithm is used, a maximum of eight bytes, in other words 16 digits, can be encrypted at one time. Thus, the combination number KOZ1 can be enlarged in the direction of the lower-value positions to include an additional value. For example, the added portion can be part of the serial number SN or the number of the advertisement frame WRN, or it can be that byte which is selected from the data block of the advertisement frame depending on an additional value.

This combination number KOZ1 can be encrypted into cryptonumber KRZ1 in substep 452 in approximately 210 ms, whereby a number of additional essentially known steps also occur at this time. Then, in substep 453, the cryptonumber KRZ1 must be converted into a corresponding row of symbols based on a predefined marking list which is stored in memory area M of nonvolatile working memory 5. In this way, an increased information density, which is so advantageous in the subsequent imprint, can be produced.

Even if a set of 10 symbols were printed - shown in Figure 3f - in other words without an increase in the information density compared to the cryptonumber KRZ1, but in two rows of marks (next to or above and beneath one another), additional symbols could remain and could be used to represent additional information,

either in encrypted or nonencrypted form. Preferably, this is information which never or hardly ever changes, and only needs to be encrypted and converted into a row of symbols one time. Preferably, this is the value G5, in other words inspection data (INS), for example the date of the last inspection, or the remainder of the serial number SN, or the SN and the byte of the data block in the advertisement frame which was not incorporated in the first combination number KOZ1, or selected in predefined parts thereof. Figure 3c shows in windows FE6 and FE10 - which are here arranged orthogonal to one another - a row containing a total of 20 symbols with which, for example, the total 8 bytes, in other words 16 digits, of the cryptonumber KRZ1 and additional information is reproduced, possibly in nonencrypted form or encrypted in some other way.

A second variant containing step 45b in addition to step 45a differs from the first variant by having different initial or input values, which are to be utilized in a similar fashion. In the second variant, a row of marking symbols is generated in two sequential steps, 45b and 45a, whereby step 45b is carried out in a manner analogous to that of step 45a.

Here, in a first substep of step 45, which is performed by the controller 6, a check is made to determine whether a flag was set for initiating the execution of steps 45b or 45a, such that in substep 45b a second combination number KOZ2 which possesses at least the other portion of the values G0, G1 which characterize the user of the postage meter is then encrypted to form a second cryptonumber KRZ2, and then converted at least into a second marking symbol row MSR2 using a second symbol set SSY2.

In substep 455, compared with substep 451, a combination number KOZ2 is generated, whereby here specifically the values for remaining portions of the serial number, for the advertisement (frame) number, and other values can be incorporated. In substep 456, as in substep 452, a cryptonumber KOZ2 is generated. Then in substep 457 the cryptonumber is transformed into a row of marking symbols, which is temporarily stored in a nonvolatile manner in substep 458.

This is followed by substeps 451 to 453, which comprise substep 45a. This may be followed by substep 454. Then point c_3 is reached.

In spite of the fact that the DES algorithm is used two times here, time is indeed saved since a first substep 450 evaluates whether the selected values which are needed to generate the marking symbol row in substep 45b have been changed by some input. In the event of a new input of selected specific values, a flag would be set in step 44 and this flag would be taken into account in the subsequent generation of data for a new row of marking symbols in order to process step 45b to completion. However, if this is not the case, then the marking symbol row which was generated earlier and which is stored in nonvolatile form in memory area 458 [sic] or parts of the marking symbol row can be reused.

In one embodiment, an encryption algorithm different from DES is used in substep 456 to save time.

In a further preferred embodiment a transformation is performed in substep 453 of the first embodiment or in substep 457 of the second embodiment to further increase the information density in the marking symbol row with respect to cryptonumbers KRZ1 or KRZ2. For example, with a cryptonumber having 16 digits, a set of 22 symbols is now used so that the information can be reproduced using only 12 digits - in the manner shown in Figure 3b. The row of marking symbols shown in this figure is doubled for two cryptonumbers. This can be accomplished by means of an additional row of marking symbols located parallel to the row of marking symbols shown in Figure 3b.

Correspondingly, it can be further shown that all that is required for a 14-digit marking symbol row is a symbol set containing 14 symbols. The test performed by the postal authority, which was described above, of articles of mail containing such rows of marking symbols can therefore in the second evaluation variant be accomplished by transforming the row of marking symbols back into cryptonumbers KRZ1 or KRZ2, which are then subsequently decrypted into combination numbers KOZ1 and KOZ2, whose individual values are compared with the values overtly printed in the imprint on the piece of mail.

A row of marking symbols such as that shown in Figure 3a is designed for 10 digits and can represent a cryptonumber KRZ1 if the symbol set contains 40 symbols. Here, it makes sense to utilize fully automated input and evaluation, if for no other reason than to avoid subjective errors by the inspector in reading the symbols.

In a step following step 45, the data from a data block for the row of marking symbols are embedded in the remaining pixel data after they have been decompressed. Specifically, two different possibilities are provided in accordance with the invention. The one possibility is illustrated by Figure 11, the other by Figure 13.

Figure 11 specifically illustrates step 46 of Figure 5. In substep 4660 the window characteristic values Z_k and T_k for modified window data are declared, the window change number p' is determined, and a window count variable q is set to zero. Substep 4661 evaluates whether window count variable q is equal to the window change number p'. If so, then point d_3 and

hence the next step 47 have already been reached. However, this path is not routinely followed initially, since the monotonically increasing values continually generate new rows of marking symbols for each imprint.

On the other hand, when a change is made, processing branches to substep 4662 in order to input window characteristic values corresponding to the changed windows and to set initial conditions.

Substep 4663 generates a new source address for the data in the data block of the window FE_k which is currently being processed so that in the next substep, 4664, one byte of coded type-2 window data can be loaded from memory area B_k into the register of nonvolatile memory 7a and so that control codes can be detected.

Substep 4665 then increments the window column run length Y_k by an amount equal to the window column run length variable W_k, which here is still zero. The program then searches for control codes for inking changes (substep 4666) and, in some cases, branches back to substep 4663 or searches for the control code for end of column (substep 4667). If this is successful, the process branches to substep 4669, and the window column counter P_k is incremented. If not, the control code must be decoded in the next substep 4668 and the fetched byte must be converted into decompressed binary type-2 window pixel data.

Substep 4670 then checks to see if all the columns of the window have been processed to completion. If so, processing branches to substep 4671 and the column run length Y_k of window FE_k is stored in the memory 7b and processing branches back to substep 4673.

If substep 4670 detects that all the columns have not yet been processed to completion, it branches back to substep 4663 via substep 4672, whereby window characteristic value Y_k and the inking flip-flop are reset to zero. The next step, 4668, then, if necessary, decodes the control code and converts the fetched byte into type-2 decompressed binary window pixel data.

Following substep 4673, in which the characteristic values of the next modified window are called, processing branches back to substep 4661. Point d_3 is reached when all the changed windows have been processed to completion.

The printing routine for the assembly of data from pixel memory areas I and II shown in Figure 12 is completed when a print request is detected in step 47 and data have been loaded in substep 471, which is not shown in Figure 5.

In substep 471, the end address Z_{end} is loaded, the running address Z (running variable) is set to the value of the source address Z_0 in area I of pixel memory 7c, the window column counter P_k is set to the value corresponding to the stored window column variable T_k, the window bit count length X_k is set to the value corresponding to the stored window column run length Y_k, and the destination addresses Z_k for k = p windows as well as the total run length R for a print column S_k are loaded. The print column has N print elements.

Then, when point e_1 is reached at the beginning of step 48, a number of substeps are executed. In substep 41, register 15 of the printer controller 14 is loaded serially bit by bit from area I of the pixel memory 7c with binary print column data which were called with address Z, and the window counter h is set to a number which corresponds to the window count p incremented by one. A window counter h is decremented in substep 482. This window counter outputs window numbers k sequentially, after which in substep 483 the address Z reached in the pixel memory is compared with the window starting address Z_k of window FE_k. If the result is positive and a window starting address has been reached, processing branches to substep 489, which in turn is comprised of substeps 4891 to 4895. If not, processing branches to substep 484.

Substep 4891 serially loads a first bit from area II of pixel memory 7c for window FE_k the binary pixel data (sic - ungrammatical sentence) into register 15, whereby in substep 4892 address Z and the bit count variable I are incremented and the window bit count length X_k is decremented. In substep 4893, if all the bits have not yet been loaded corresponding to the window column run length Y_k, additional bits are loaded from area II. If all the bits have been loaded, then processing branches to substep 4894, whereby the window starting address Z_k for addressing the next window column is increased by an amount equal to the total length R, and the window column counter P_k is decremented. At the same time, the original window bit count length X_k is restored corresponding to the window column run length Y_k.

Substep 4895 then checks whether all the window columns have bene processed to completion. If this is the case, then the starting address $Z_{\bar{k}}$ for the corresponding window FE_k is set to zero or an address is set which lies outside pixel memory area I. If the window columns have not processed to completion, and following substep 4896, processing branches to point e_1.

Substep 484 checks whether all window starting addresses have been scanned. If so, then processing branches to substep 485 in order to increment the running address Z. If this has not yet been done, processing branches back to substep 481 in order to decrement the window counter h until the next window

starting address is found or until in substep 484 window counter h is equal to zero.

Substep 486 checks whether all the data for the column which is to be printed s_k are loaded into register 15. If not, then the bit count variable 1 is incremented in substep 488 in order to return to point e_1 and to then (in substep 481) load the next bit, which is addressed with address Z, from the pixel memory area into register 15.

However, if register 15 is full, then the column is printed out in substep 487. Then in step 50 - which is already represented in Figure 5 - a check is made to determine whether all the pixel data in pixel memory areas I and II have been printed out, in other words whether the stamping of the piece of mail is completed. If so, then point f_1 is reached. Otherwise, processing branches to substep 501 and bit count variable 1 is reset to zero, so that processing can then branch back to point e_1. The next print column can now be generated.

The print routine for assembling the data taken from only one pixel memory area I and from working memories will be illustrated in greater detail based on Figure 13. After a print request, which is detected in step 47 and shown in Figure 6, substep 471 follows immediately. This substep has already been described in conjunction with Figure 12. This brings processing to point e_2. Step 49 - which was already represented in Figure 6 - now begins. It comprises substeps 491 to 497 and substeps 4990 to 4999. Substeps 491 to 497 execute with the same results and in the same order as substeps 481 to 487, which have already been explained in conjunction with Figure 12. Substep 493 branches to substep 4990 in order to reset an inking flip-flop to g = 0 whereupon the operation of column-by-column decompression of the coded type-2 window data is begun with substep 4991. This procedure was already explained in conjunction with Figure 6. An inking change occurs - as already explained in conjunction with Figure 7 - in the evaluation of the type-2 window pixel data which are to be converted, so that the first hexadecimal data from the called data block are to be evaluated, for example, as inked. The source address is incremented. Then, the compressed type-2 window data for window FE_k, in particular for the marking data, are loaded from the predefined data block (stored in the corresponding submemory areas B_j) into register 200 of volatile working memory area 7a in substep 4992. A hexadecimal number "QQ" corresponds to one byte.

During this process, the control code is also detected. If a window column which begins with non-inked pixels - in other words with pixels which are not to be printed - is to be printed, the first position in the data block at this point would contain the control code "inking change." Substep 4993 would then branch back to substep 4991 to perform the inking change. Otherwise, processing branches to substep 4994. Substep 4994 determines whether a control code for the "end of column" is present. If not, then the contents of the register must be decoded and, thus, decompressed. For each run length-encoded hexadecimal numerical value there exists in the character memory (BEEN) 9 a series of binary pixel data which can be called based on the hexadecimal number loaded in the volatile working memory 7a. This is done in substep 4995, whereby the decompressed type-2 window pixel data for a column of window FEj are then loaded serially into the print register 15 of the printer controller 14.

In substep 4996 the addresses are then incremented and a corresponding next hexadecimal number is selected in the data block which is stored in nonvolatile working memory 5 in subarea B_5, the bits which were converted in the decoding of the run length coding are identified in order to generate a window column run length variable W_j, with which the destination address is incremented. The new address which is to be read in is thus generated, and processing can branch back to substep 4991.

Once the end of the column is reached, substeps 4997 to 4999 can follow, so that processing can finally branch back to point e_2. Substeps 4998 and 4999 execute in a manner similar to that of substeps 4895 and 4894, which are shown in Figure 12.

The fully loaded print column is printed in substep 497. Substeps 491 to 497 execute in a manner similar to that of substeps 481 to 487, which are shown in Figure 12.

In addition to achieving a simplicity of mechanical components and lower expense, a high printing speed is achieved when a multitude of variable print image data are embedded in a stored nonchanging print image.

The preferred embodiments in particular have been illustrated in relatively great detail, although with faster hardware it would definitely be possible to modify the order of the process steps and still quickly produce a security imprint.

After a print request has been made, step 47 waits for step 48, which contains a print routine. If a print request has not been made, step 47 goes into a wait loop to wait for the print request by looping back directly to the beginning of step 47 - as shown in Figures 5 and 6. This offers a further time-related advantage, since it means that the DES algorithm does not have to be constantly regenerated. The next processable time after the marking symbol row has

been generated can already trigger printing, even though, as already noted, other branch-backs are possible.

Likewise, in a different embodiment, step 45 can be placed between steps 53 and 54. In step 54, which follows step 45, the data from a data block for the marking symbol row are embedded, after they have been decompressed, in the remaining pixel data of pixel memory area I. Thus, an additional pixel memory area is not necessary.

A contrasting embodiment stores only the frame pixel data in the pixel memory area and embeds all the window pixel data immediately in the corresponding columns read into the print register 15, so that a pixel memory does not need to be used temporarily for window data.

In one embodiment, which does not provide for the editing of advertisement texts, memory area A_i can be eliminated. Instead, the image information which does not change is stored in a read-only memory, for example in the program memory (ROM) 11. This read-only memory 11 is accessed when the nonchanging image information is decoded, so that the temporary memory can be eliminated.

The invention is not limited to the present embodiment. Rather, a number of variants utilizing the described solution to the problem are conceivable, even when the embodiments are fundamentally different.

Claims

1. A process for rapid generation of a security mark with provision of values to print these values in encrypted form in the indicium in addition to the postage and date stamp,
characterized by
a substep (45a) or a step (45) performed by a microprocessor of the controller (6) of the franking machine before a print request (step 47), comprising the substeps:
   a) generation of a combination number (KOZ1), whereby a constantly monotonically variable value (G4) to generate first related digits and at least one additional value (G3) characterizing the piece of mail to generate second related digits of the combination number (KOZ1) are made available,
   b) encryption of the combination number (KOZ1) into a cryptonumber (KRZ1),
   c) conversion of the cryptonumber (KRZ1) into at least one marking symbol series (MSR1) using a set (SSY1) of symbols.

2. The process according to Claim 1, characterized in that the constantly monotonically variable value is at least one machine parameter increasing or decreasing during the service life of the franking machine, in particular a time count or complement thereof.

3. The process according to Claims 1 and 2, characterized in that a machine parameter is time-dependent and includes a value (G4a), characterizing the decreasing battery voltage of the battery-supported memory, and a second constantly monotonically decreasing value (G4b) or the respective complements of the values (G4a and G4b).

4. The process according to Claims 1 through 3, characterized in that the second constantly monotonically decreasing value (G4b) is the complement of the mail item count or is a constantly monotonically decreasing time-dependent value.

5. The process according to Claims 1 and 2, characterized in that the constantly monotonically decreasing value is a counting value corresponding to the next inspection date (INS) and is a constantly monotonically decreasing time-dependent value.

6. The process according to Claims 1 and 2, characterized in that a constantly monotonically increasing value includes the date or the mail item count determined at the last inspection.

7. The process according to Claims 1 through 6, characterized in that a part of a value (G0, G1) characterizing the user of the franking machine is made available by the controller (6) to generate third related digits of the combination number (KOZ1).

8. The process according to Claims 1 through 7, characterized in that in a first substep (450) of the step (45) performed by the controller (6), a check is made, as to whether a flag has been set to trigger the performance of substeps (45b) and/or (45a),
that in the substep (45b) a second combination number (KOZ2) with at least the other part of the value (G0, G1) characterizing the user of the franking machine is generated, then encrypted into a second cryptonumber (KRZ2), and finally converted into at least one second marking symbol series (MSR2) using a set (SSY2) of

symbols.

9. The process according to Claims 1 through 8, characterized in that the generation of cryptonumbers (KRZ1 and/or KRZ2) is performed by the controller (6) of the franking machine by implementation of a DES algorithm, for which a single key or appropriate different keys (KEY1 or KEY2) are stored in the franking machine.

10. The process according to one of Claims 1 through 9, characterized in that before the conversion of the cryptonumber a transformation is performed in such a counting system by an algorithm before the conversion of the cryptonumber, which [counting system] permits a higher density of data by an expanded set (SSY2) and/or (SSY1) of symbols for at least one marking symbol series (MSR1).

11. The process according to Claims 1 through 10, characterized in that the symbols have only orthogonal edges, which are, on the one hand, machine readable, in particular in the form of an integral light/dark evaluation, for checking by the postal authorities, and that, on the other hand, they enable manual evaluation, in particular through their symbolism, whereby in each case a predefined meaning is allocated to the symbols. [Tr. note: Claim 11 is grammatically incorrect in German. There is a singular verb with no singular subject. The claim was modified somewhat in English to try to make sense of it based on the text of the patent.]

12. The process according to Claims 1 through 11, characterized in
   a) that other steps to edit the window data of Type 1, which do not change so frequently as the window data of Type 1 [sic], and to load registers from the memory areas B_J precede the step (45), in order to load at least one pixel storage area I and/or II with decompressed pixel data or to reload new window pixel data.
   b) that after the steps (450 through 453) of the substep (45a) performed to generate a marking symbol series or after the steps (455 through 457) of the substep (45b), the step (45) includes an additional step (454) or (458) to generate at least one data record for window data of Type 2 and possibly to store it in the memory areas B_K, whereby the aforementioned changeable window data of Type 2 are changeable data which make each security mark differentiable and thus unmistakable and are provided for at least one window (FE6, FE10) in the indicium.

13. A process for rapid generation of a security mark with control of the column by column printing of an indicium in a franking machine with an electronic printer, whereby variable data stored encoded in memories and constant data of an image record are processed separately before printing and then combined into one printed image,
characterized in
   - that after the franking machine is turned on, the bytes of at least a first data record, which includes run-length coded, constant, hexadecimal data of a printing format, in particular frame data and control data, are taken (step 42) by a controller (6) with a microprocessor from a first memory (11) and/or a second nonvolatile memory (5) and temporarily stored (step 43) in a register (100) of a first volatile memory (7a),
   - that during the aforementioned step (43), control data are detected by the microprocessor controller (6) from the encoded bytes of the first data record and possibly window characteristics (Z, T, Y) are generated for at least one window area and temporarily stored in a second volatile memory (7b), and that binary pixel data from the frame data are decoded and temporarily stored in a third volatile memory (7c),
   as well as
   - that in the aforementioned step (43), the window data are moved from the second nonvolatile memory (5) in binary pixel data decoded in the third volatile memory (7c) and then for execution of a print routine in step (48) column by column, bit by bit into a print register (14) of a printer controller (15); or that in step (49), window data are moved directly column by column, bit by bit into the print register (14) of a printer controller (15), whereby the transfer of the window pixel data occurs sequentially with the frame pixel data.

14. The process according to Claim 13, characterized in that after a print request, the pixel data are taken in step (47) by the controller (6) from the third volatile memory (7c), whereby the second

volatile memory (7b) is polled for window characteristics.

15. The process according to Claim 14, characterized in that the window characteristics of the first and of at least one second data record are used by the controller (6) to integrate the transferred decoded binary window pixel data in a pixel storage area II of the third volatile memory (7c) into the binary pixel data of the respective print columns in a step (48) for execution of the print routine and to store them in the print register (14). (Variant 1b)

16. The process according to Claim 14, characterized in that the window characteristics of the first and of at least one second data record are used by the controller (6) to integrate the window data stored in the second nonvolatile memory (5), after their decoding into binary window pixel data, bit by bit into the binary pixel data of the respective print columns in a step (48) for execution of the print routine and to store them directly in the print register (14). (Variant 1a)

17. The process according to one of the preceding Claims 13 or 14, characterized in that the window data of Type 1 or Type 2 are decompressed along with the frame data before printing and are loaded into the pixel storage area I of the third volatile memory (7c) and that upon a print request the print data are moved into the print register (14) from the third volatile memory (7c) by the controller (6), sequentially in a manner suited to column by column printing. (Variant 2)

18. The process according to one of the preceding Claims 13 through 17, characterized in
   - that run-length coded, constant, hexadecimal data, in particular data of a selected indicium including frame data and control data are removed from the first nonvolatile memory (PSP 2) of the franking machine and stored in the second nonvolatile memory (5) of the franking machine, byte by byte in the form of a first data record in a first area A_i, and
   - that, on the other hand, separate therefrom, run-length coded, variable, hexadecimal data, of the indicium, which correspond to the current settings of the data for the indicium window stored in storage areas T of the second nonvolatile memory (5) are stored byte by byte in the form of a second data record, including in particular window data and control data, in a second area B_i.

19. The process according to Claims 13 through 18 characterized in that [the system] pauses in the performance of the print routine (step 48 or 49 until the print request.

20. The process according to Claim 19, characterized in that [the system] pauses in the performance of the print routine (step 48 or 49) until the print request, while [the system] branches to a step (44) or (45).

21. The process according to Claim 20, characterized in that upon reentering of predefined special values, a flag is set and is polled in step (45).

22. A process for rapid generation of a security mark, characterized by a combination of the characteristics of Claims 1 through 12 with Claims 13 through 15 and 18 through 21, whereby the window pixel data stored in a pixel storage area II are generated from a third data record for window data of Type 2 in a step (46), which [data record] is stored separately after first and second data records in an additional area B_K, whereby the third data record contains run-length coded, variable, hexadecimal marking data generated in the step (45) for an unmistakable mark.

23. A process for rapid generation of a security mark, characterized by a combination of the characteristics of Claims 1 through 12 with the characteristics of Claims 13 and 14, 16, and 18 through 21.

24. A process for rapid generation of a security mark, characterized by a combination of the characteristics of Claims 1 through 12 with the characteristics of Claims 13 and 17 through 21.

25. An arrangement for rapid generation of a security mark for franking machines with a printer module for a fully electronically generated indicium with at least one input means, one display means, one input/output control module, one nonvolatile memory for at least the constant parts of the indicium as well as with a controller and with a printer controller, which generates the print design, which has been composed from fixed data and current data by the microprocessor-controlled print process,

characterized in that
the data for the constant parts of the indicium involve at least the frame of an advertising block and are stored in a first storage area A_i and an allocated name identifies the frame of the block, that the data for the variable parts of the indicium are stored in a second storage area B_j and an allocated name identifies the variable part, that the data for a first allocation of names of the variable parts are present in a third storage area C.

26. The arrangement according to Claim 25, characterized in that for automatic modification of text in the block, names of the block text sections are allocated to the name of at least one constant part, in particular the advertising block selected, and are present stored in a predefined manner in the third storage area C of the nonvolatile memory (5), and that the selected block frame is in each case allocated to a cost center KST and/or a second allocation of block types to cost centers is present stored in a fourth storage area D.

27. The arrangement according to Claims 25 and 26, characterized in that the names of the block frames stored in a first storage area A_i are composed of the number of the cost center KN and a type number TN, that the names of the block text sections, which are stored in a second storage area B_s, are determined by the current date and possibly by the time of day, and that the data for the first allocation in the third storage area C are determined by storage both of the name of the block frame containing the cost center number KN and the type number TN as well as of the name of the block text section containing the current date and possibly the time of day.

28. The arrangement according to Claims 25 through 27, characterized in that the display unit (3) presents only the names of those block frames for which allocations falling within a defined period of time exist in the third storage area C.

29. The arrangement according to Claims 25 through 28, characterized in that the defined period of time is automatically reported in relationship to the current date through the components of the name of the allocation of block text sections, in particular the month and/or the year.

30. The arrangement according to Claim 29, characterized in that an allocation is present in a fifth storage area E which [allocation] involves a freely programmable defined period of time, in which the names of the allocations of block text sections to block frames are associated with the current date.

31. The arrangement according to one of Claims 25 through 30, characterized in that data for the generation of additional barcode window data from a combination number or values or a cryptonumber are present in a sixth storage area F.

32. The arrangement according to one of Claims 25 through 31, characterized in that input means (2) are provided to store the data measured or determined during an inspection — e.g., the total number of frankings, the next inspection date, etc. — in a seventh storage area G.

33. The arrangement according to one of Claims 23 through 32, characterized in that a cryptonumber (KRZ2) can be temporarily stored in an eighth storage area K.

34. The arrangement according to one of Claims 25 through 33, characterized in that at least one set (SSY) for marking symbols is stored in a ninth storage area M.

35. The arrangement according to one of Claims 25 through 34, characterized in that at least one key (KEY1, KEY2) for an encryption algorithm is stored in a tenth storage area S on the one hand, and the encryption algorithm itself is also stored there.

36. The arrangement according to one of Claims 25 through 35, characterized in that the data with number strings for all input values are stored in an eleventh storage area T.

37. The arrangement according to one of Claims 25 through 36, characterized in that a compression algorithm converts the BCD-packed number representation for the base ten into a differently packed number representation for a base greater than ten in order to represent at least one cryptonumber (KRZ1) with one marking symbol series (MSR1) with data compression.

38. The arrangement according to one of Claims 25 through 37, characterized in that security means are provided for coordination with the input means (3), which [security means] enable access to the storage areas G, S, M, and V at the time of an inspection, such that at the time of an inspection a new key (KEY), encryption algorithm, marking symbol set (SSY), or compression algorithm can be loaded via the input means (2) and that these security means are present in the form of a physical key and/or a chip card.

39. The process according to Claim 12, characterized by the following steps preceding process steps (51, 52) for modification of block text sections for franking machines:
   a) loading of agreed types of blocks via modem or chip card,
   b) selection of a block frame,
   c) editing of a block text section in the franking machine,
   d) combination and display of a total representation of the indicium.

40. The process according to Claim 39, characterized in that before editing of a block text section in the franking machine, there is a display of a number of names of the block frames for selection of a block frame or an optical display for selection of the block frame with allocated block text sections or standard text sections from a pixel memory, and by acknowledgment of the selection an allocation of at least one of the variable text sections to each block frame was made; that by means of the control elements, editing of the block text section selected takes place with simultaneous optical presentation of the text on the display, as well as that after the editing of a block text section and the combination and the display of a total representation of an indicium, the edited text section is automatically stored before franking.

41. The process according to Claims 12 and 39 characterized in that the representation of the names of block text sections in the window provided for this occurs in that field of the display unit in which an optical display occurs after the selection, or that the optical displays appear for selection in the window of the field of the display unit one after another in an order determined by the time components of the name.

42. A process for verification of the security mark, characterized by the steps
   a) visual detection of the serial number and its input via an input device (25),
   b) visual detection of the postage value and its input via the input device (25),
   c) visual detection of the graphic symbols and their input via an input device (25) with appropriately identified function keys,
   d) starting of an automatic evaluation, possibly in cooperation with a data center (21) and signaling of the results of the comparison or display of at least part of the values regenerated from the mark for manual verification by an examiner of the postal authority.

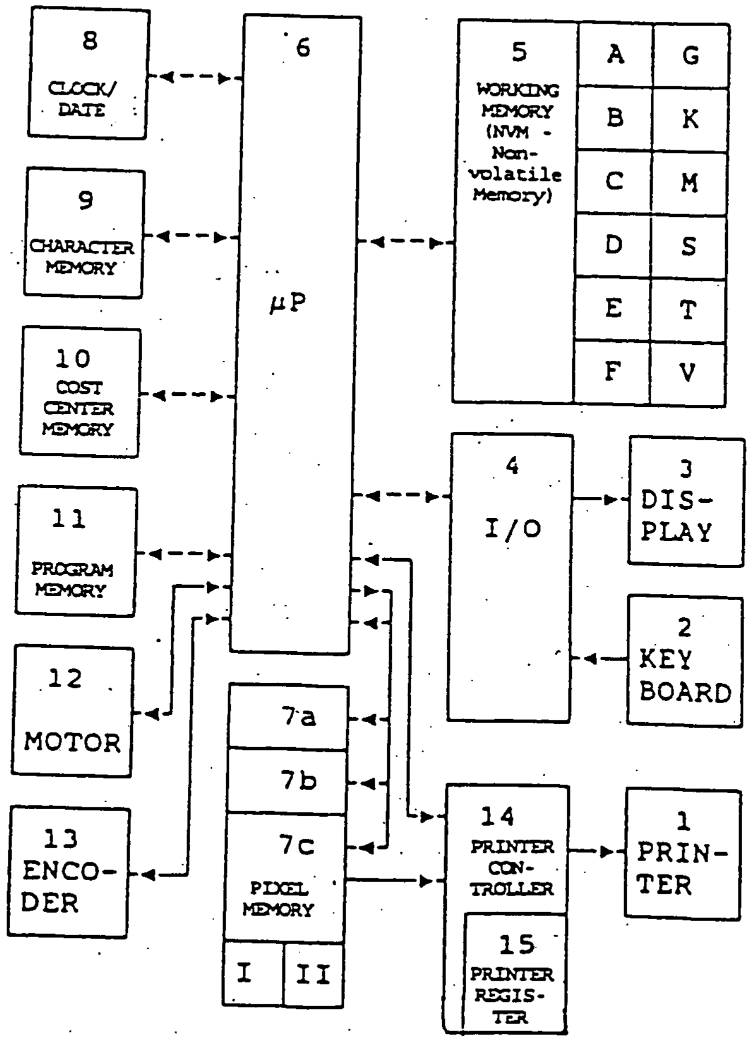

8
CLOCK/
DATE
9
CHARACTER
MEMORY
10
COST
CENTER
MEMORY
11
PROGRAM
MEMORY
12
MOTOR
13
ENCO-
DER
6
μP
7a
7b
7c
PIXEL
MEMORY
I
II
5
WORKING
MEMORY
(NVM -
Non-
volatile
Memory)
A
G
B
K
C
M
D
S
E
T
F
V
4
I/O
3
DIS-
PLAY
2
KEY
BOARD
14
PRINTER
CON-
TROLLER
15
PRINTER
REGIS-
TER
1
PRIN-
TER

Fig. 1

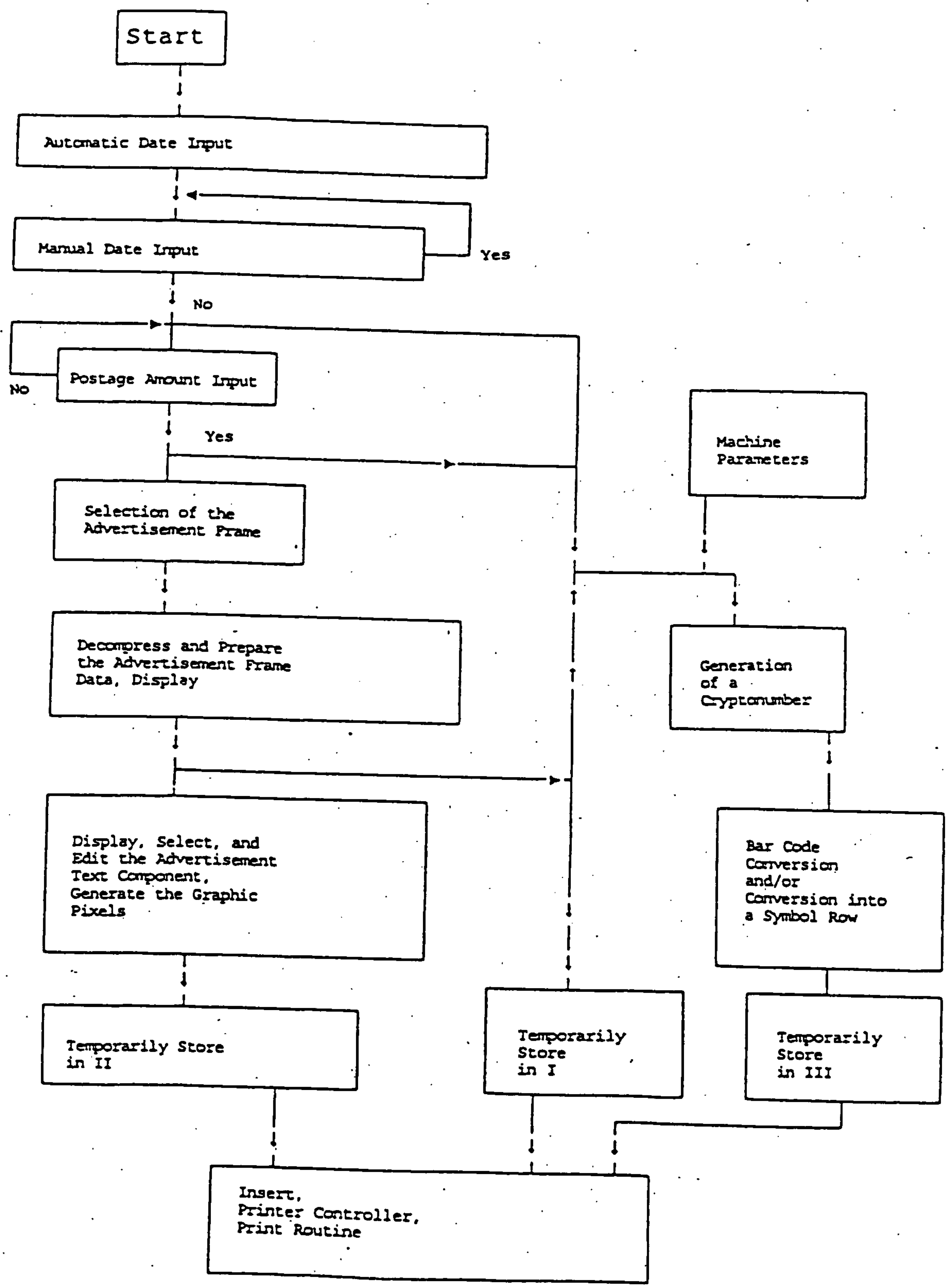

Start
Automatic Date Input
Manual Date Input
Yes
No
No
Postage Amount Input
Yes
Machine Parameters
Selection of the Advertisement Frame
Decompress and Prepare the Advertisement Frame Data, Display
Generation of a Cryptonumber
Display, Select, and Edit the Advertisement Text Component, Generate the Graphic Pixels
Bar Code Conversion and/or Conversion into a Symbol Row
Temporarily Store in II
Temporarily Store in I
Temporarily Store in III
Insert, Printer Controller, Print Routine

Fig. 2

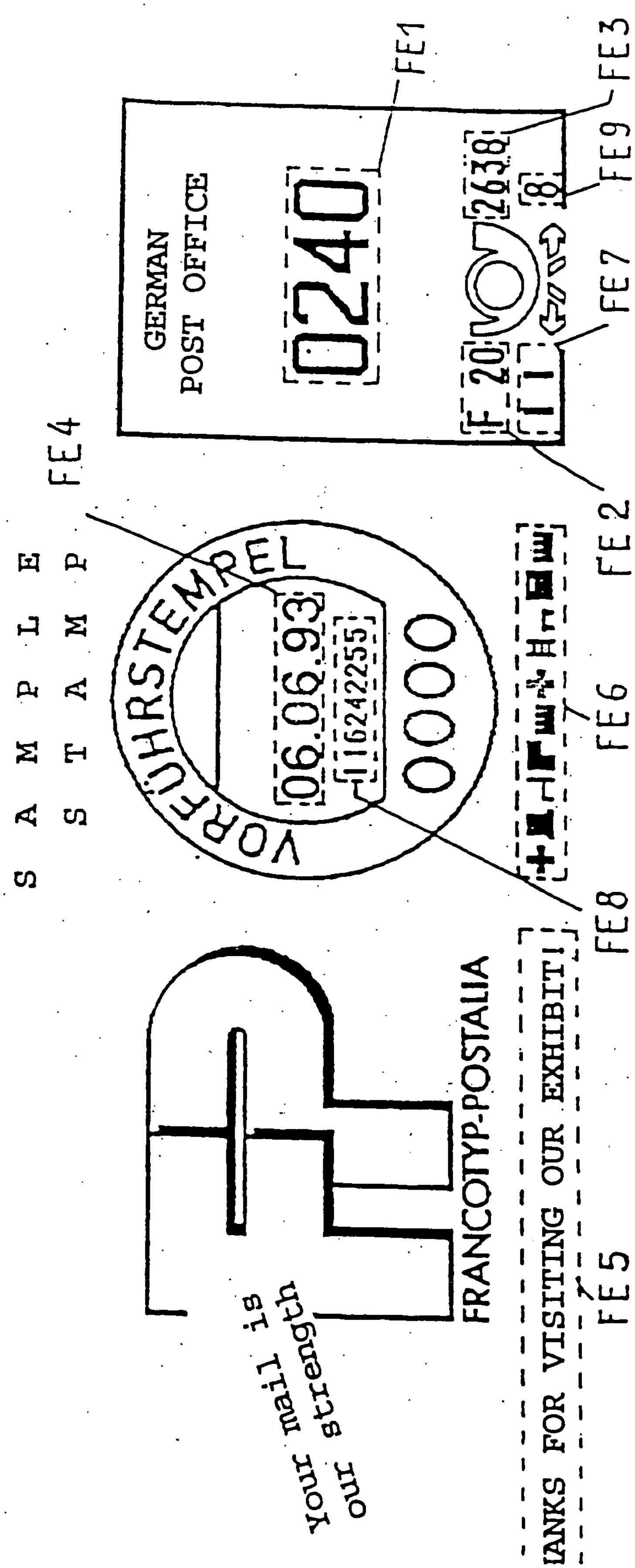

GERMAN
POST OFFICE
0240
FE1
2638
FE3
8
FE9
FE7
F 20
FE2
SAMPLE
STAMP
VORFÜHRSTEMPEL
06.06.93
116242255
0000
FE4
FE6
FE8
FRANCOTYP-POSTALIA
Your mail is
our strength
ANKS FOR VISITING OUR EXHIBIT!
FE5

Fig. 3a

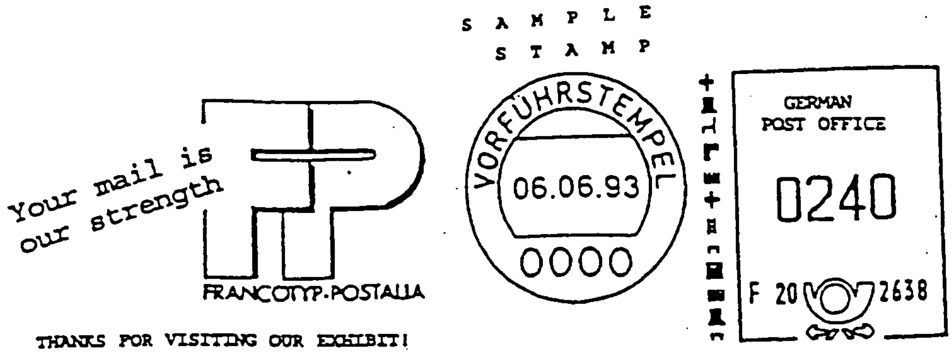


Fig. 3b

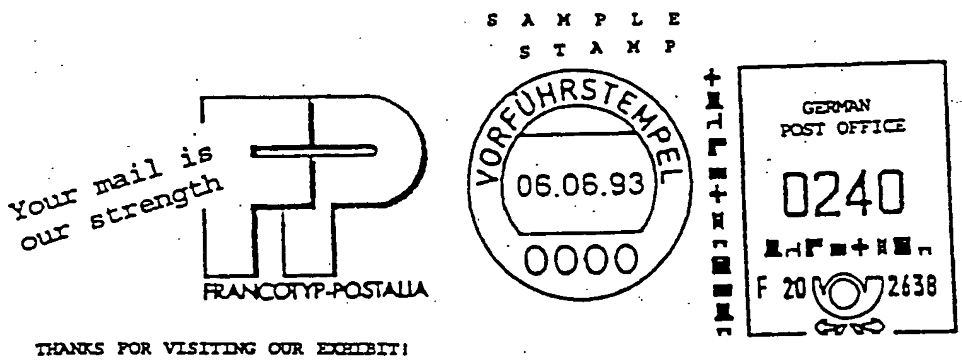


Fig. 3c

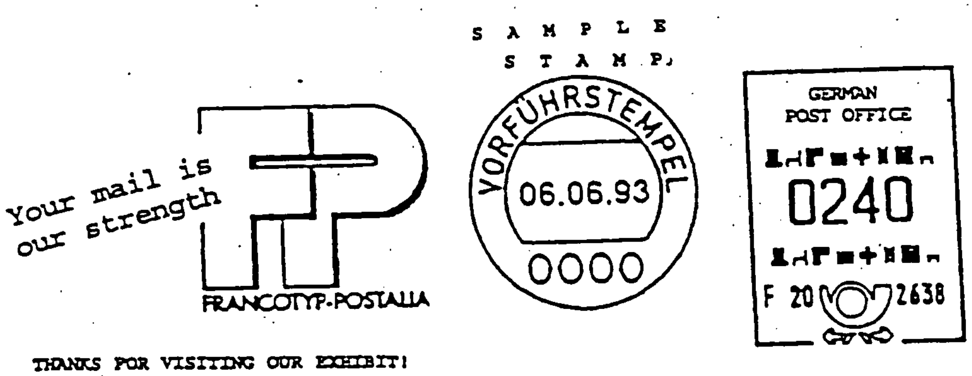
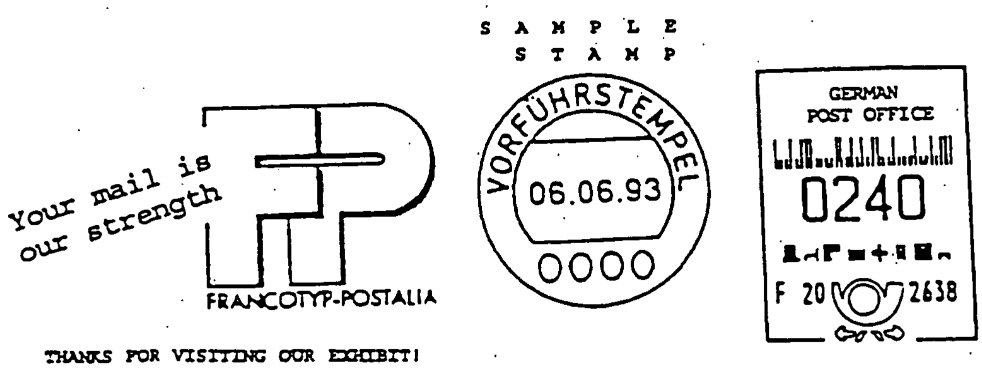
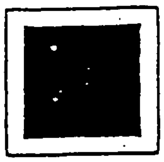
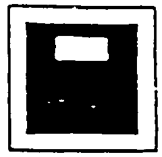


Fig. 3d

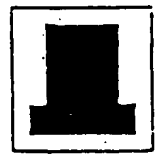




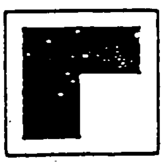
1. "Square" Symbol
Density: 100%



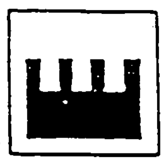
2. "Door" Symbol
Density: 90.1%



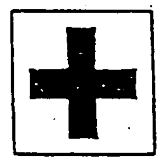
3. "Hat" Symbol
Density: 79.6%



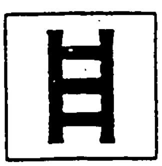
4. "Angle" Symbol
Density: 71.4%



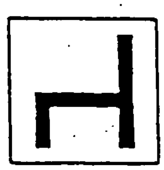
5. "Crown" Symbol
Density: 59.2%



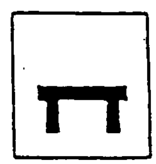
6. "Cross" Symbol
Density: 49.0%



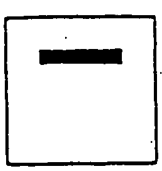
7. "Ladder" Symbol
Density: 40.8%



8. "Chair" Symbol
Density: 29.6%



9. "Table" Symbol
Density: 20.4%



10. "Bar" Symbol
Density: 10.2%

Fig. 3f

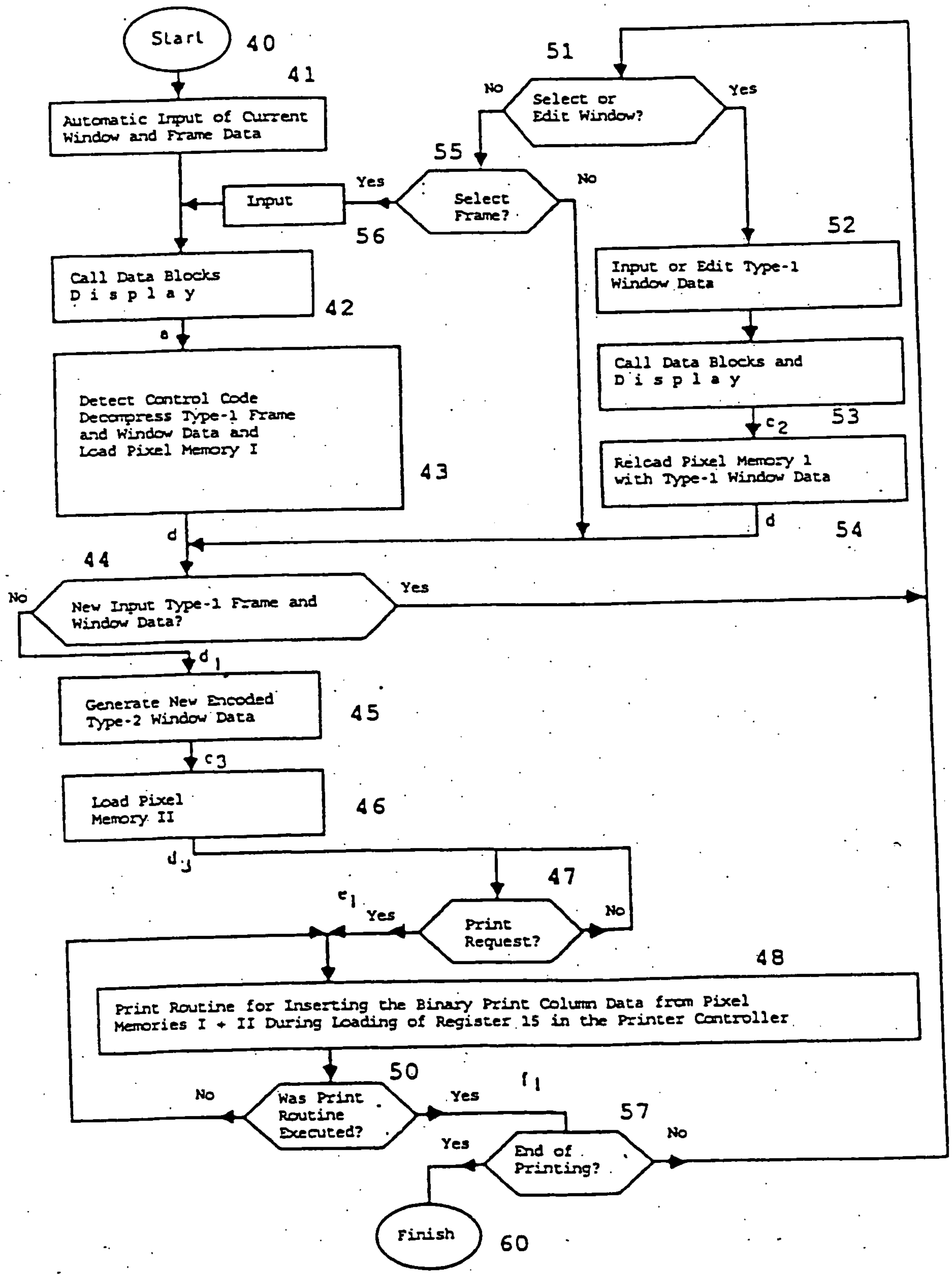
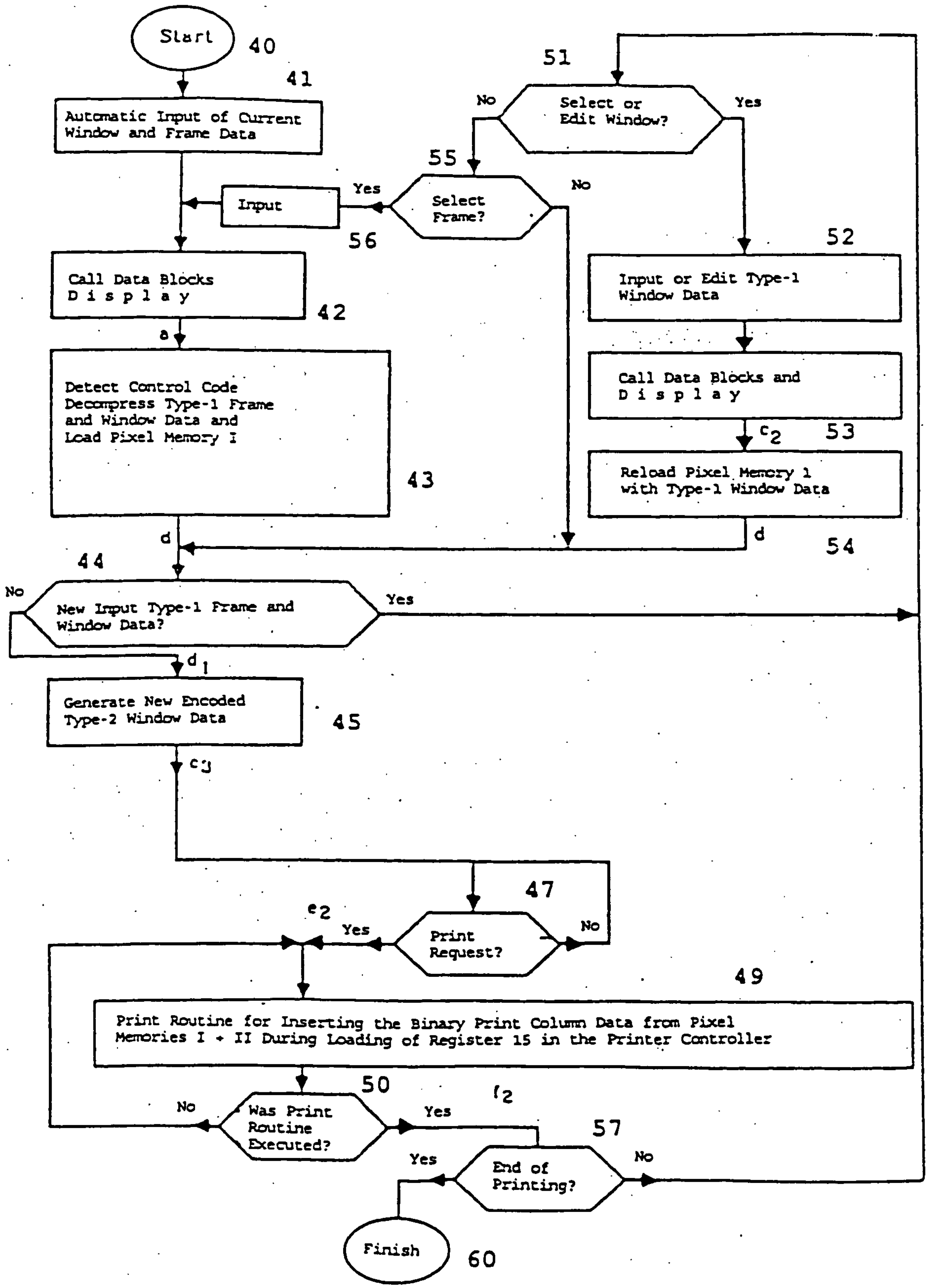

Start
40
41
Automatic Input of Current Window and Frame Data
51
Select or Edit Window?
No
Yes
55
Select Frame?
Yes
No
Input
56
52
Input or Edit Type-1 Window Data
Call Data Blocks Display
42
a
Call Data Blocks and Display
c2
53
Detect Control Code Decompress Type-1 Frame and Window Data and Load Pixel Memory I
43
Reload Pixel Memory 1 with Type-1 Window Data
d
54
d
44
No
New Input Type-1 Frame and Window Data?
Yes
d1
Generate New Encoded Type-2 Window Data
45
c3
Load Pixel Memory II
46
d3
47
e1
Yes
Print Request?
No
48
Print Routine for Inserting the Binary Print Column Data from Pixel Memories I + II During Loading of Register 15 in the Printer Controller
50
f1
No
Was Print Routine Executed?
Yes
57
Yes
End of Printing?
No
Finish
60

Fig. 5



Start
40
41
Automatic Input of Current Window and Frame Data
51
Select or Edit Window?
No
Yes
55
Select Frame?
Yes
No
Input
56
Call Data Blocks Display
42
52
Input or Edit Type-1 Window Data
Call Data Blocks and Display
c_2
53
Reload Pixel Memory 1 with Type-1 Window Data
a
Detect Control Code Decompress Type-1 Frame and Window Data and Load Pixel Memory I
43
d
d
54
44
New Input Type-1 Frame and Window Data?
No
Yes
d_1
Generate New Encoded Type-2 Window Data
45
c_3
47
e_2
Print Request?
Yes
No
49
Print Routine for Inserting the Binary Print Column Data from Pixel Memories I + II During Loading of Register 15 in the Printer Controller
50
Was Print Routine Executed?
No
Yes
f_2
57
End of Printing?
Yes
No
Finish
60

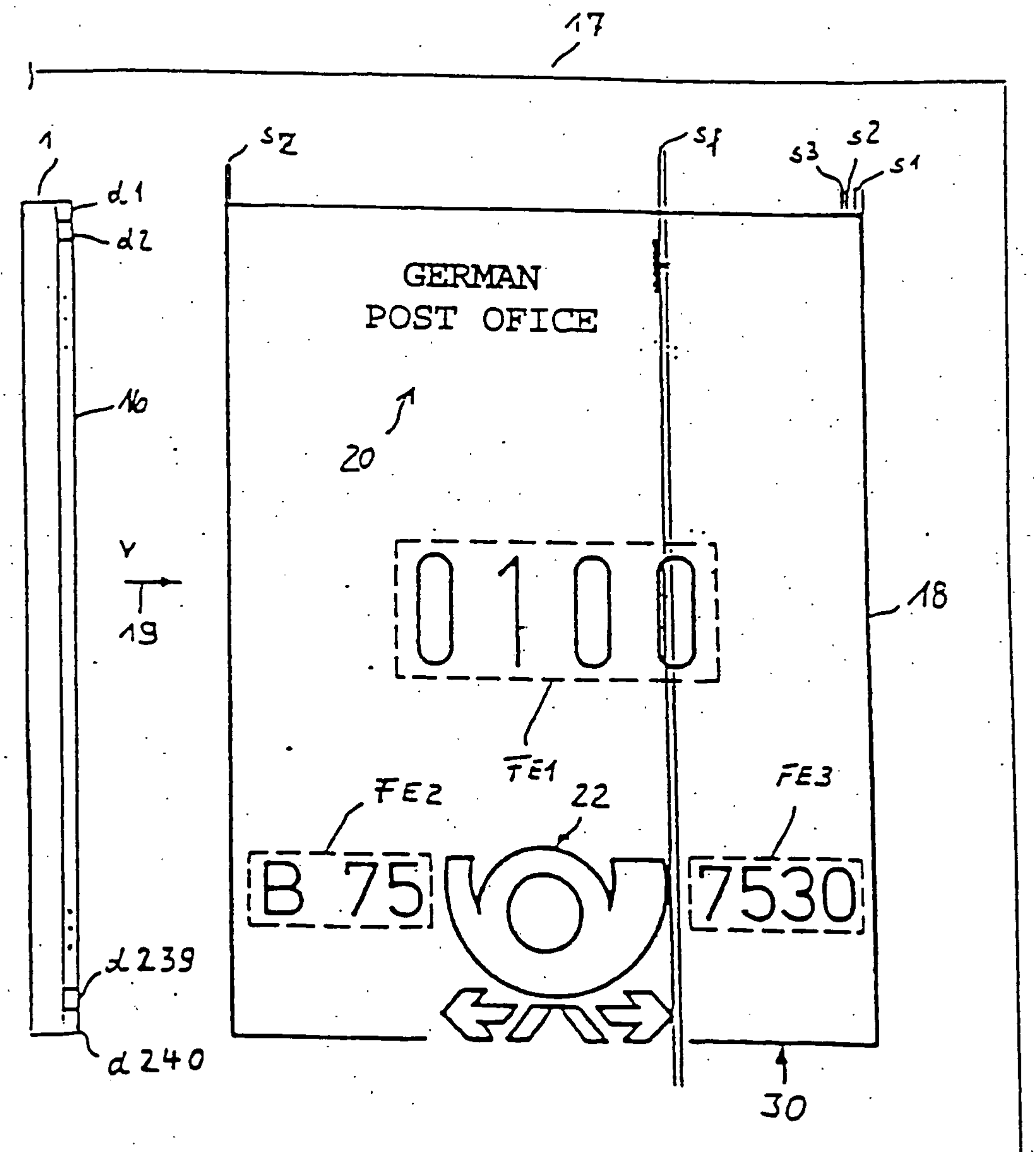

17
1
d1
d2
16
d239
d240
V
19
s2
s1
s3
s2
s1
GERMAN
POST OFICE
20
0100
FE1
18
FE2
22
FE3
B 75
7530
30

Fig. 7

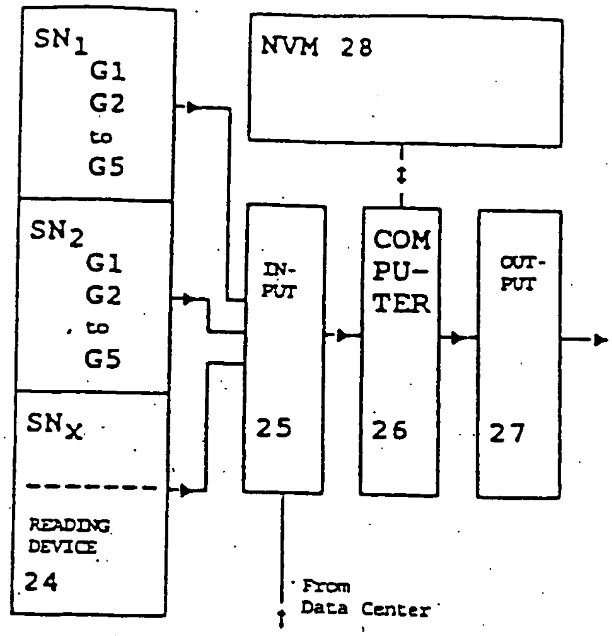
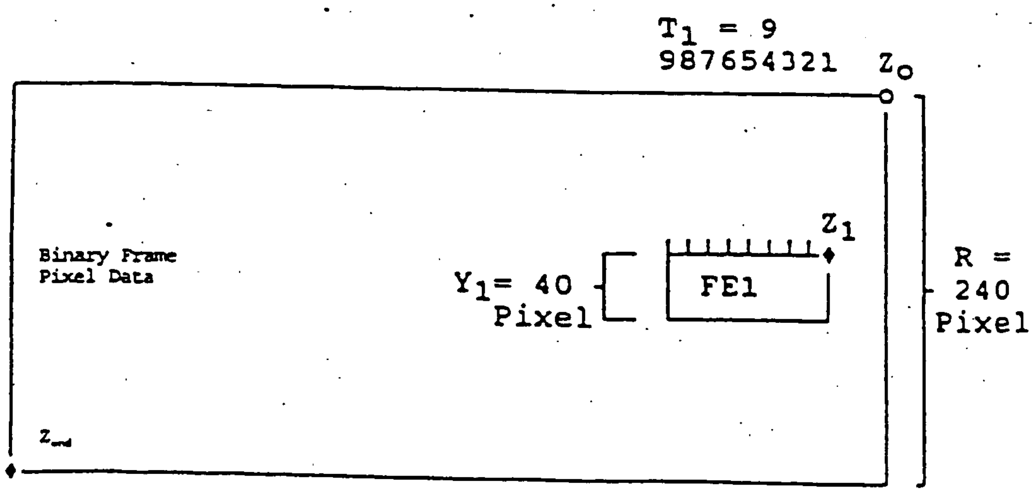


Fig. 4



Z1: Destination Address for RAM 7c in RAM 7b for 1st Window

T1: Window Column Variable in RAM 7b for 1st Window

Y1: Window Column Run length of 1st Window = Constant

Fig. 8. Window characteristic values for window no. j = 1 in relation to the binary pixel data.

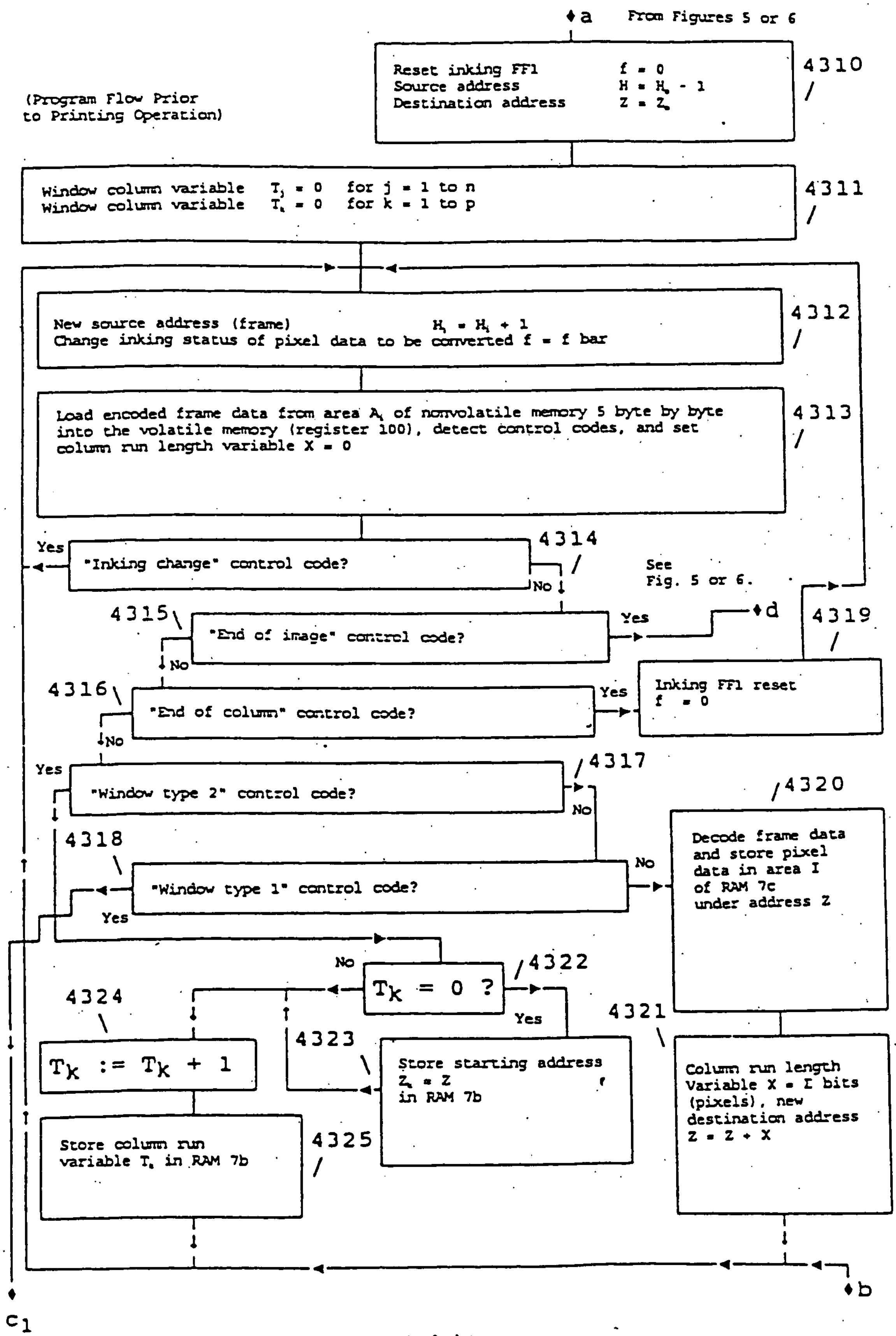


Fig. 9a. Step 43a: Load the frame pixel data into pixel memory area I.

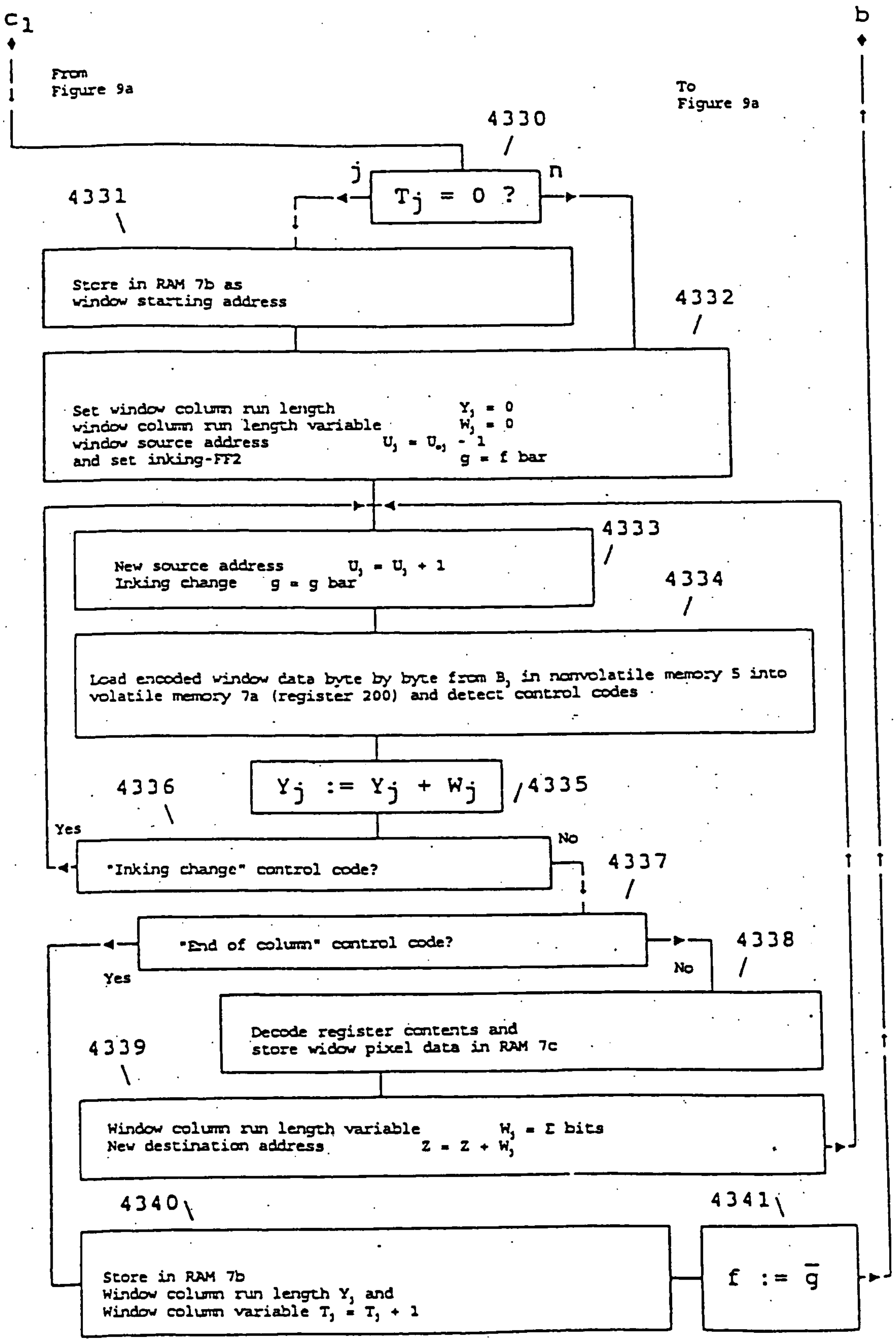
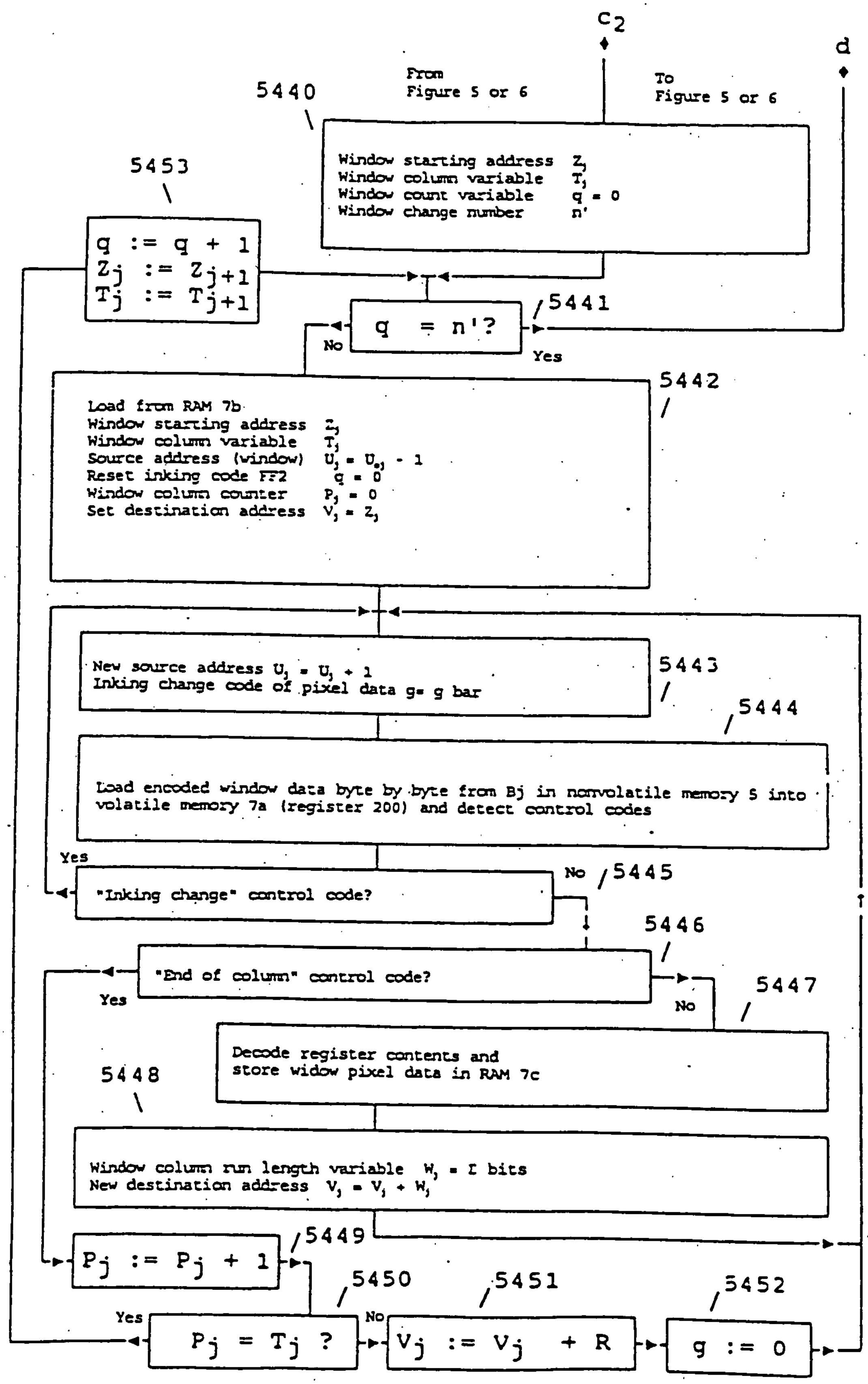


Fig. 9b. Step 43b: Load the "type-1" window pixel data


c2
d
From
Figure 5 or 6
To
Figure 5 or 6
5440
Window starting address Z_j
Window column variable T_j
Window count variable $q = 0$
Window change number n'
5453
$q := q + 1$
$Z_j := Z_{j+1}$
$T_j := T_{j+1}$
$q = n'?$
5441
No
Yes
5442
Load from RAM 7b
Window starting address Z_j
Window column variable T_j
Source address (window) $U_j = U_{oj} - 1$
Reset inking code FF2 $q = 0$
Window column counter $P_j = 0$
Set destination address $V_j = Z_j$
New source address $U_j = U_j + 1$
Inking change code of pixel data g= g bar
5443
5444
Load encoded window data byte by byte from Bj in nonvolatile memory 5 into volatile memory 7a (register 200) and detect control codes
Yes
"Inking change" control code?
No
5445
5446
"End of column" control code?
Yes
No
5447
Decode register contents and store widow pixel data in RAM 7c
5448
Window column run length variable $W_j = \Sigma$ bits
New destination address $V_j = V_j + W_j$
$P_j := P_j + 1$
5449
5450
5451
5452
Yes
$P_j = T_j$?
No
$V_j := V_j + R$
$g := 0$

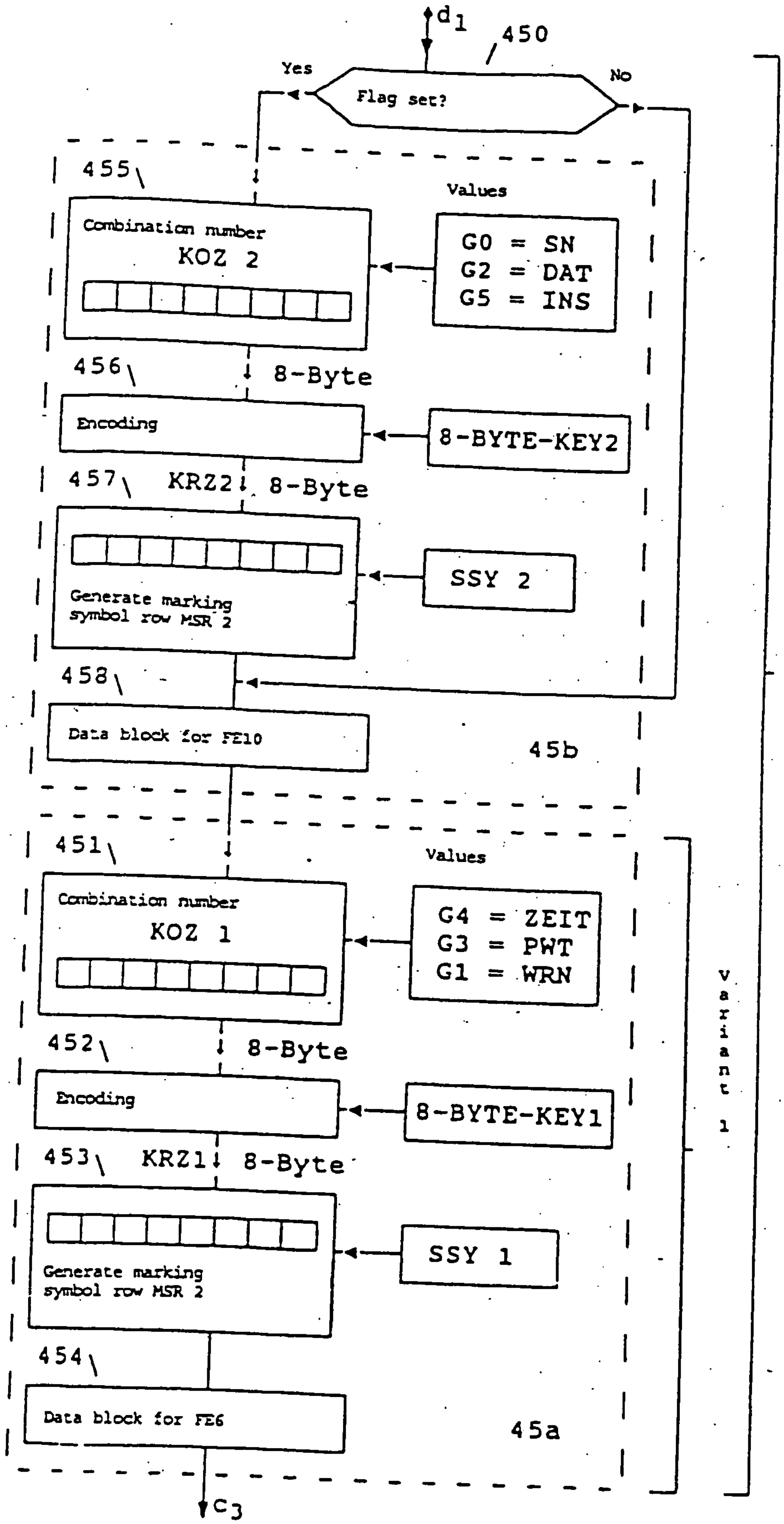
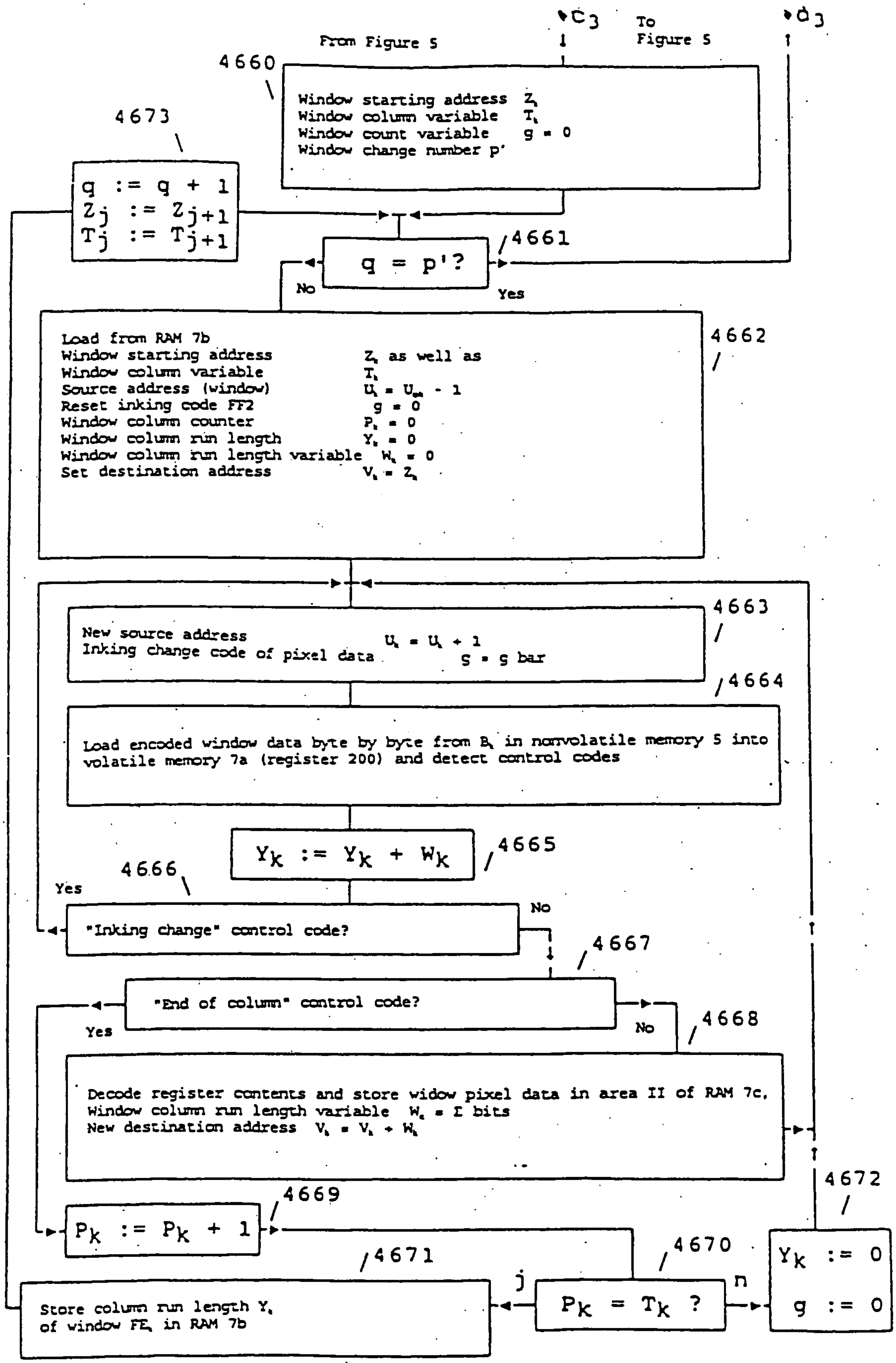


Fig. 10. Step 45:



c3
To Figure 5
d3
From Figure 5
4660
Window starting address Z
Window column variable T
Window count variable g = 0
Window change number p'
4673
q := q + 1
Zj := Zj+1
Tj := Tj+1
4661
q = p'?
No
Yes
4662
Load from RAM 7b
Window starting address Z as well as
Window column variable T
Source address (window) U = U - 1
Reset inking code FF2 g = 0
Window column counter P = 0
Window column run length Y = 0
Window column run length variable W = 0
Set destination address V = Z
4663
New source address U = U + 1
Inking change code of pixel data g = g bar
4664
Load encoded window data byte by byte from B in nonvolatile memory 5 into volatile memory 7a (register 200) and detect control codes
Yk := Yk + Wk
4665
4666
Yes
"Inking change" control code?
No
4667
"End of column" control code?
Yes
No
4668
Decode register contents and store widow pixel data in area II of RAM 7c,
Window column run length variable W = Σ bits
New destination address V = V + W
4669
Pk := Pk + 1
4672
4671
Store column run length Y of window FE in RAM 7b
j
4670
Pk = Tk ?
n
Yk := 0
g := 0

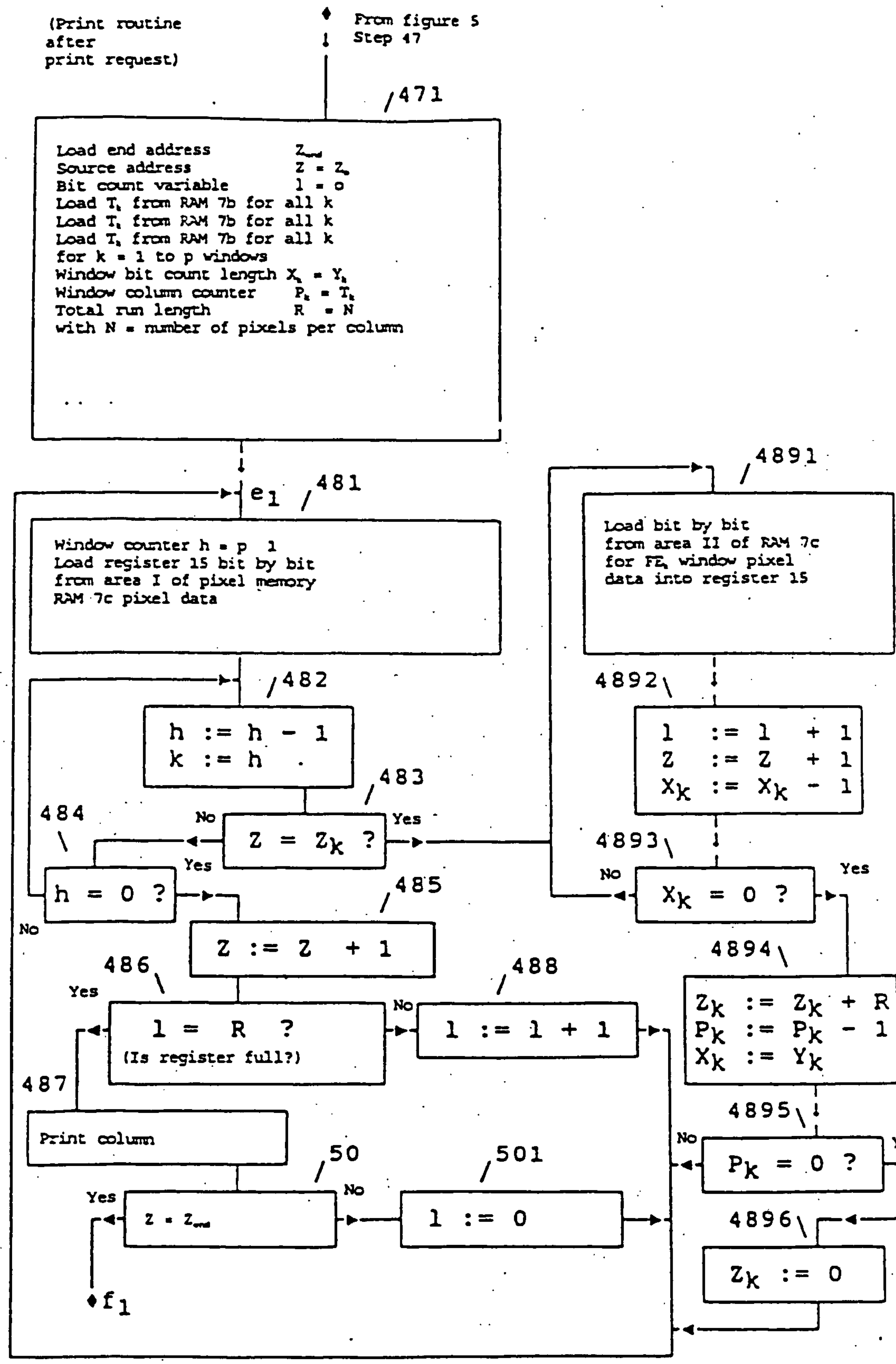


Fig. 12. Print routine (steps 48 and 50).